VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

STATISTICKÉ METODY DETEKCE ANOMÁLIÍ DATOVÉ KOMUNIKACE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

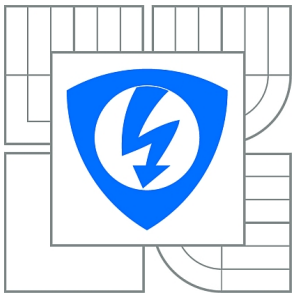
AUTOR PRÁCE
AUTHOR

Bc. EDUARD WOIDIG

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

STATISTICKÉ METODY DETEKCE ANOMÁLIÍ DATOVÉ KOMUNIKACE

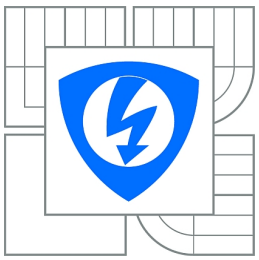
STATISTICAL ANOMALY DETECTION METHODS OF DATA COMMUNICATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. EDUARD WOIDIG
AUTHOR

VEDOUCÍ PRÁCE — Mgr. KAREL SLAVÍČEK, Ph.D.
SUPERVISOR

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Eduard Woidig ***ID:*** 165848
Ročník: 2 ***Akademický rok:*** 2014/2015

NÁZEV TÉMATU:

Statistické metody detekce anomálií datové komunikace

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je prozkoumat možnosti využití statistických metod, zejména analýzy časových řad, pro detekci anomálií v datové komunikaci a to zejména bezpečnostních útoků.
Úkolem je prozkoumat vliv anomálií na kvantitativní parametry datové komunikace (průměrná délka paketu, poměr TCP/UDP/dalších typů komunikace atd.).

DOPORUČENÁ LITERATURA:

[1] ENDORF, Carl. Detekce a prevence počítačového útoku. 1. vyd. Praha: Grada, 2005, 355 s. ISBN 80-247-1035-8.
[2] ARORA, Himanshu. TCP Attacks: TCP Sequence Number Prediction and TCP Reset Attacks. [online]. Dostupné z: http://www.thegeekstuff.com/2012/01/tcp-sequence-number-attacks/

Termín zadání: 9.2.2015 ***Termín odevzdání:*** 26.5.2015

Vedoucí práce: Mgr. Karel Slavíček, Ph.D.
Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Tato práce slouží jako teoretický základ pro praktické řešení problematiky použití statistických metod pro detekci anomálií v datovém provozu. Základní zaměření detekcí anomálií datového provozu je na datové útoky. Proto hlavní náplní je analýza datových útoků. V rámci řešení jsou datové útoky řazeny dle protokolů, které útočníci ke své činnosti zneužívají. V každé části je popsán samotný protokol, jeho využití a chování. Pro každý protokol je postupně řešen popis jednotlivých útoků, včetně metodiky vedení útoku a postihů na napadený systém nebo stanice. Dále jsou pro nejzávažnější útoky nastíněny postupy jejich detekce a případné možnosti obrany proti nim. Tyto poznatky jsou shrnuty do teoretické analýzy, která by měla sloužit jako výchozí bod pro praktickou část, kterou bude samotná analýza reálného datového provozu.

Praktická část je rozdělena do několika oddílů. První z nich popisuje postupy pro získávání a přípravu vzorků tak, aby na nich bylo možné provést další analýzy. Dále jsou zde popsány vytvořené skripty, které slouží pro získávání potřebných dat ze zaznamenaných vzorků. Tato data jsou detailně analyzována za použití statistických metod, jako jsou časové řady a popisná statistika. Následně jsou získané vlastnosti a sledovaná chování ověřována za pomocí uměle vytvořených i reálných útoků, kterými je původní čistý provoz modifikován. Pomocí nové analýzy jsou modifikované provozy porovnány s původními vzorky a provedeno vyhodnocení, zda se podařilo nějaký druh anomálie detekovat.

Získané výsledky a sledování jsou souhrnně shrnuty a vyhodnoceny v samostatné kapitole s popisem dalších možných útoků, které nebyly přímou součástí testovací analýzy.

KLÍČOVÁ SLOVA

detekce, anomálie, útok, záplava, zneužití, protokol

ABSTRACT

This thesis serves as a theoretical basis for a practical solution to the issue of the use of statistical methods for detecting anomalies in data traffic. The basic focus of anomaly detection data traffic is on the data attacks. Therefore, the main focus is the analysis of data attacks. Within the solving are data attacks sorted by protocols that attackers exploit for their own activities. Each section describes the protocol itself, its usage and behavior. For each protocol is gradually solved description of the attacks, including the methodology leading to the attack and penalties on an already compromised system or station. For the most serious attacks are outlined procedures for the detection and the potential defenses against them. These findings are summarized in the theoretical analysis, which should serve as a starting point for the practical part, which will be the analysis of real data traffic.

The practical part is divided into several sections. The first of these describes the procedures for obtaining and preparing the samples to allow them to carry out further analysis. Further described herein are created scripts that are used for obtaining needed data from the recorded samples. These data are were analyzed in detail, using statistical methods such as time series and descriptive statistics. Subsequently acquired properties and monitored behavior is verified using artificial and real attacks, which is the original clean operation modified. Using a new analysis of the modified traffics compared with the original samples and an evaluation of whether it has been some kind of anomaly detected.

The results and tracking are collectively summarized and evaluated in a separate chapter with a description of possible further attacks, which were not directly part of the test analysis.

KEYWORDS

detection, anomaly, attack, flood, exploit, protocol

WOIDIG, Eduard *Statistické metody detekce anomálií datové komunikace*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 76 s. Vedoucí práce byl Mgr. Karel Slavíček, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Statistické metody detekce anomálií datové komunikace“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Mgr. Karlu Slavíčkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

ÚVOD

Anomálie v datovém provozu mohou mít vícero příčin. Ne každý výkyv vůči standardnímu provozu nemusí hned znamenat datový útok. Může se jednat o nárůst provozu související s nějakou marketingovou kampaní nebo významnou událostí. V poslední době výrazně stoupá obliba streamovaných on-line přenosů a nejvýraznější nárůst bývá v období konání sportovní akcí jako např. olympijské hry nebo šampionáty v hokeji či fotbale. Tyto nárůsty však bývají časově omezené a po ukončení dané akce by se datový provoz měl vrátit opět k normálu.

Další a mnohem frekventovanější anomálie v datovém provozu jsou způsobeny a způsobovány různými druhy datových útoků. Z pohledu detekce jsou nejvýraznější výkyvy způsobeny útoky typu DoS a DDoS. Jedná se o útoky na dostupnost, případně distribuované útoky na dostupnost. Jejich účelem je znemožnit komunikaci buď vyřazením určitého spoje nebo bodu na síti nebo zahlcením sítě, takže není možné zpracovávat požadavky ostatních uživatelů. Zásadní rozdíl mezi oběma typy je, zda je útok veden z jedné nebo více stanic, tzv. botnetu.

Nejrozšířenějším nástrojem pro detekci datových útoků jsou systémy IDS – Intrusion Detection System neboli systém pro odhalení průniku. Jedná se o obranný systém, který monitoruje síťový provoz a snaží se vyhledávat a odhalovat aktivity, které by mohly být součástí datového útoku. Tyto systémy provádějí monitoring a detekci na základě předem stanovených popisů chování. Takže aby takovýto systém byl účinný, je třeba, aby byl správně nakonfigurován vůči síti, ve které se nachází, a kterou má chránit.

Detekce anomálií, která bude řešena v rámci této práce, se nedá se systémy IDS srovnávat, protože data, která jsou sbírána a budou analyzována, pochází přímo z páteřní sítě ve správě Cesnetu. Samotná detekce tedy nebude cílená na konkrétní útoky, ale bude zaměřená na anomálie v obrovském datovém provozu. Případné anomálie se dle chování pokusíme k nějakému konkrétnímu útoku přirovnat. Proto největší část teoretického řešení bude zaměřena na analýzu datových útoků.

1 DATOVÉ ÚTOKY

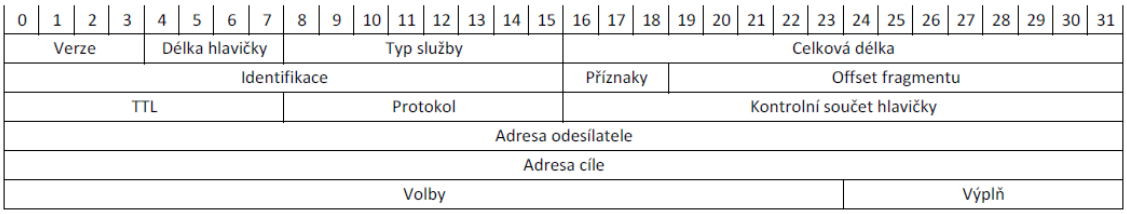
V této části jsou popsány nejčastější a nejzávažnější datové útoky včetně metodiky jejich průběhů.

1.1 Zneužití IP protokolu

Internet protokol (dále jen IP) je základní protokol pracující na síťové vrstvě. Zodpovídá za směrování paketů ze zdrojové stanice k cílovému hostiteli přes jednu nebo více IP sítí. Pokud během přenosu dojde ke ztrátě, tak IP sám o sobě neřeší jeho opětovné doručení. IP v doručování datagramů tedy poskytuje nespolehlivou službu, která může být označována jako best effort – všechny stroje na trase se datagram snaží podle svých možností poslat blíže k cíli, ale nezaručují prakticky nic. Datagram vůbec nemusí dorazit nebo může být doručen několikrát a neručí se ani za pořadí doručených paketů. Toto mají na starosti protokoly vyšších vrstev, typicky protokol TCP.

1.1.1 Podvržení IP – IP spoofing

Jak bylo zmíněno v úvodu této kapitoly, tak IP zodpovídá za transport a dodávku paketů, neposkytuje však mechanizmus, jak ověřit, že pakety odeslané z určitého uzlu byly opravdu z daného uzlu odeslány. Toho je zneužíváno při útoku IP spoofing, známého také jako podvrhování IP adresy. Útočník manipuluje s hlavičkami IP paketů (obr. 1.1), přičemž se nelegálně vydává za jiný stroj a tím získává neoprávněný přístup k počítačům.



Obr. 1.1: Hlavička IP datagramu.

Existuje několik variant útoků, které úspěšně využívají IP spoofing. I když některé z nich jsou již poměrně zastaralé, jiné jsou stále velmi aktuální a je třeba se před nimi mít na pozoru.

- Blind Spoofing – podvržení naslepo: útočník zaznamená navázané TCP spojení mezi důvěryhodnými stanicemi. Při útoku zašle oběti paket, v jehož hlavičce bude jeho IP adresa jako odesílatele nahrazena IP adresou důvěryhodného zdroje, s kterým oběť komunikuje. Tento paket je směrovači v síti bez potíží předán cílové stanici, která jej přijme. TCP vrstva příjemce paket zpracuje a následně zahodí, protože daný paket je brán jako odpověď na požadavek, který nikdy neposlala. Zpětná odpověď je směrována na IP odesílatele a k útočníkovi nedorazí, proto se tento typ útoku nazývá útokem naslepo. Takže útočník musí vědět, co bylo posláno a zároveň předpokládat, jaká bude reakce zdrojové stanice. V minulosti mohl správně vytvořený útok předat do systému oběti požadavek na vytvoření nového uživatelského účtu, což útočníkovi umožnilo vydávat se za důvěryhodné hostitele a tím získat plný přístup.
- Non-Blind Spoofing: tento typ útoku lze provést, pokud je útočník ve stejné podsíti jako oběť, aby mohl odchytit a odposlechnout pořadí a potvrzování paketů. Tento typ útoku však patří spíše mezi útoky na protokol TCP, který bude zmíněn později v kapitole 1.3.3.
  Oba výše uvedené typy útoků lze zařadit do skupiny Man-in-the-middle útoků, neboť útočník zachycuje legitimní komunikaci mezi dvěma přátelskými stranami a díky podvrhování paketů nebo hlaviček paketů může řídit tok komunikace. Tím může odstranit nebo změnit informace zaslané jedním z účastníků spojení bez vědomí původního odesílatele nebo příjemce. Navíc při Non-blind útoku přepokládá oklamaná oběť, že komunikuje s legitimním účastníkem a útočník tak dokáže získat důvěrné informace.
- IP záplavy – jedná se o DoS útoky, které jsou jedním z nejzásadnějších a nejhůře detekovatelných útoků. Při takovém útoku začne útočník nebo síť ovládnutých počítačů generovat ohromný provoz paketů s podvrženými zdrojovými IP adresami na konkrétní oběť nebo síť. Cílem útoku je zaslat co největší počet paketů v co nejkratší době, aby došlo k zahlcení příjemce a jeho vyřazení z provozu. Běžnými ochrannými mechanizmy nelze tomuto typu útoku zabránit, protože zablokování IP adres náhodně generovaných v podvržených paketech je nereálné a tím je znemožněno vysledování zdroje útoku.[16]

1.1.2 Zneužití IP fragmenatace

V rámci sítě má každá část sítě stanovenu MTU (Maximum Transmission Unit, maximální přenosová jednotka), která určuje maximální velikost datagramu, který je možné přenést. Když je přenášený datagram příliš velký a není ho možné přenést v rámci jednoho IP datagramu, je potřeba jej rozdělit na fragmenty.
Pro řízení fragmentace jsou v hlavičce IP datagramu(obr. 1.1) položky:

- Identifikace – identifikace daného fragmentu.
- Offset fragment – pozice fragmentu v originálním paketu.
- Příznak Don't Fragment – bitový příznak označující zákaz fragmentace.
- Příznak More Fragments – bitový příznak zda následují další fragmenty.

Fragmentace IP nastane, když směrovač v síti obdrží paket větší než MTU (Maximum Transmission Unit) příštího segmentu sítě. Pokud paket nemá bitový příznak DF, tedy zákaz fragmentace, může směrovač datagram fragmentovat dle MTU následující trasy. Všechny tyto fragmenty budou mít v poli identifikace stejnou hodnotu a v poli fragment offset bude uvedena pozice aktuálního fragmentu v původním paketu. V cílové stanici dojde ke spojení všech fragmentů do původního datagramu, který je následně předán do vyšších vrstev. Samotné sestavení je možné uskutečnit až po doručení všech fragmentů.[12]

Sestavování fragmentů však není tak přímočaré, jak by se zdálo, a stává se cílem útoků.

- Teardrop útok – útok, který využívá fragmentace paktů a vytváří překrývající se pakety. V implementaci IP vrstvy téměř všech operačních systémů jsou chyby v kódu, případně nedostatečně kódované algoritmy, které zajišťují zpětné sestavení datagramů. Útočník může vytvořit a odeslat pár IP paketů, ve kterých speciálně upravil offset fragmentů. Při procesu zpětného skládání mohou server zmást, případně jej donutit až k pádu. Další možností je, že původní datagram se spojí, ale velmi překvapujícím způsobem. Předpokládejme, že první fragment byl 24 bajtů dlouhý, ale druhý fragment tvrdí posun pouze o 20 bajtů. Při zpětném sestavení tak data v druhém fragmentu přepíšou poslední čtyři bajty dat prvního fragmentu, čímž dojde k jeho znetvoření.
- Vyhýbání se firewallu – známé také jako útok krátkými fragmenty. Při tomto typu útoku útočníci vytvoří uměle fragmentované pakety s cílem zahltit nebo vyřadit firewall z provozu. Pro rozhodnutí firewallu, jak s paketem naložit, potřebuje kompletní informaci, tedy zdrojovou a cílovou IP adresu a porty. Fragmenty, které mu dorazí, jsou tak krátké, že tyto informace neobsahují a firewall si je musí sám skládat, čímž vynakládá vlastní prostředky.
- Pakety liché délky – délka fragmentů vyjma poslední části by měla být dělitelná 8. Pokud ostatní fragmenty toto nesplňují, měly by být velmi podezřelé, protože se pravděpodobně jedná o podvrh.
- Při skládání fragmentů je vypočítávána délka následujícího fragmentu. U některých útočníkem upravených fragmentů může přijímací hostitel vypočítat, že druhý fragment má zápornou délku. Pokud je tato informace předána do

paměti, tak je záporné číslo převedeno na extrémně dlouhé kladné číslo, což zapříčiní selhání segmentace.

- Fragmenty lze vytvořit tak, aby při zpětném spojení vznikl mimořádně velký paket, větší, než je maximální povolená délka IP paketu. Nejznámější druh tohoto typu útoku je Ping smrti – Ping of Death. Bude zmíněn později v kapitole 1.4.3 o zneužití protokolu ICMP.

1.1.3 Detekce a prevence zneužití IP

Detekce IP spoofingu vyžaduje koordinované úsilí mnoha počítačů a bezpečnostních vlastností jejich operačních systémů. Jedním z možných řešení je průběžně aktualizovaná databáze strojů, které v daném okamžiku nejsou on-line. Další možností je, že si všechna jádra řídících systémů jednotlivých zařízení vedou detailní záznamy o požadavcích na výměnu IP adresy v záznamech za případnou podvrženou IP adresu. O této změně poté informují ostatní důvěryhodné systémy.

Detailní detekci lze také provést dle vlastníků jednotlivých IP adres a adresních rozsahů. Datagram, který je odesílán na externího síťového poskytovatele a má jako zdrojovou a cílovou IP adresu uvedeny adresy v místní doméně je s největší pravděpodobností podvržený. Dalším způsobem, jak zjistit IP spoofing, je sledování a porovnávání procesů jednotlivých účtů připojených mezi systémy vnitřní a vnější sítě. V případě, že se útok s podvrženou IP podařil, by měl být na počítači oběti v logu záznam ukazující vzdálený přístup, avšak na zdrojovém počítači, který měl tento přístup provést, nebude v logu existovat žádná odpovídající položka o zahájení tohoto přístupu.

Základní prevencí proti IP spoofingu je využití speciálně upravených systémů. Jádro takovéhoto systému může odmítnout předat ethernetový rámec dál sítí, když zdrojová IP adresa neodpovídá adrese přiřazené k danému poskytovateli, od něhož rámec přichází. Zároveň je možné prověřovat, zda odesílající hostitel pracuje na důvěryhodném systému. Tuto akci je možné v rámci zabezpečení přidat i přesto, že TCP oficiálně nevyžaduje ověření hostitele žádajícího o spojení.

Další možností, jak výrazně zvýšit obranu proti útokům pomocí podvržení IP adresy, je filtrování příchozího a odchozího datového provozu na hraničním routeru. Do filtrovacích pravidel je třeba přidat ACL (Access Control List, seznam pro řízení přístupu), který na vstupním rozhraní blokuje privátní IP adresy. Navíc by toto rozhraní nemělo přijímat pakety se zdrojovou IP adresou, která je z rozsahu vnitřní sítě, protože to je spoofing technika používaná k obcházení firewallů. Na odchozím

rozhraní je vhodné zakázat odesílání paketů se zdrojovou adresou mimo vnitřní rozsah sítě, aby se zabránilo vytváření falešného provozu někomu ve vnitřní síti.

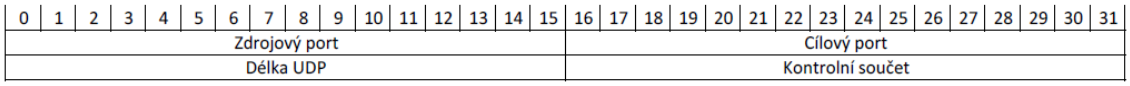
Hrozby IP spoofing útoku rovněž snižuje provádění šifrování a autentizace. Obě tyto funkce jsou zahrnuty v protokolu IPv6, který bude lépe eliminovat stávající spoofing hrozby.

Pro detekci zneužití fragmentace IP paketů je třeba využít speciálních detekčních systémů.

1.2 Zneužití UDP protokolu

UDP – User Datagram Protokol je protokol transportní vrstvy. Poskytuje jednoduché rozhraní mezi síťovou a aplikační vrstvou. Oproti tomu však neposkytuje žádné záruky doručení. Protokol UDP je nespojovaná služba, tedy nenavazuje spojení. Odesílatel odešle UDP datagram příjemci a už se nestará o to, zda se datagram náhodou neztratil. O to se musí postarat protokol aplikační vrstvy. Jeho hlavička (viz obrázek 1.2), je více než jednoduchá, skládá se pouze z 4 polí, z čehož 2 jsou volitelná.

- Zdrojový port – port aplikace, která datagram vytváří, může být prázdný.
- Cílový port – port aplikace, které má být datagram předán.
- Kontrolní součet – není povinný, může být 0.



Obr. 1.2: Hlavička UDP protokolu.

1.2.1 Zneužití kontrolního součtu UDP

Protože vyplňování pole kontrolní součet v hlavičce UDP datagramu není povinné, neexistuje u datagramů s prázdným polem způsob, jak detekovat porušení během přenosu. Protože UDP provoz je velmi rozšířený, tak většina systémů toto pole využívá i za cenu drobné ztráty výkonnosti. Důvodem je výrazné zvýšení spolehlivosti. UDP provoz, u kterého je pole kontrolní součet prázdné, bude automaticky podezříván, že se vyhýbá detekci.

Nejedná se tedy přímo o zneužití, ale je to možnost, jak se vyhnout bližším kontrolám a pokusit se nějaký typ útoku provést.

1.2.2 Záplava UDP – UDP flood

Jedná se o druh DoS útoku, ve kterém útočník zaplavuje náhodné porty na cílové stanici IP pakety, které obsahují UDP datagramy. Příjemce zkontroluje aplikace spojené s těmito porty datagramů a odešle zpět ICMP zprávu cíl nedosažitelný. Čím více UDP paketů je přijímáno a odpovědí odesíláno, tím více se systém stává ohromen a přestává reagovat na ostatní příchozí požadavky. V rámci útoku zaplavením UDP využívá útočník obvykle ještě techniku IP spoofing, čím si zajistí, že návratové ICMP pakety ho nezasáhnou a zároveň skrývá svou identitu jako zdroje útoku.[3]

Jelikož UDP provoz nevyžaduje trojcestný handshake jako TCP spojení, běží s nižšími režijními náklady a je tedy ideální pro provoz, který nemusí být tolik kontrolován. Nicméně tyto vlastnosti činí UDP náchylnější ke zneužití. Vzhledem k tomu, že při navazování spojení chybí počáteční handshake a rovnou se vytváří platné spojení, může útočník okamžitě rozesílat ohromný provoz přes UDP kanály, aniž by ho cokoliv při odesílání omezovalo. To znamená, že záplavové UDP útoky jsou nejen vysoce efektivní, ale také, že mohou být provedeny s využitím poměrně málo prostředků. Pokud má útočník k dispozici síť ovládnutých počítačů, stává se jeho útok ještě silnějším.

1.2.3 Detekce a prevence zneužití UDP

Jak již bylo zmíněno výše, UDP je nespolehlivý protokol. Datagramy v UDP aplikacích mohou přicházet mimo pořadí nebo bez předchozího upozornění dorazí některé chybějící datagramy. Protože se neprovádí kontrola doručení jednotlivých datagramů, je UDP rychlejší a efektivnější pro mnoho aplikací. Pro detekci tohoto typu útoku neexistuje v současné době žádný konkrétní nástroj, který by jej spolehlivě odhalil. Všechny detekční systémy pracují na principu sledování statistických hodnot datového provozu a detekci anomálií.

UDP provoz obecně využívá malou část přenosové kapacity, proto mohou náhlé změny přenášeného UDP provozu být obecně označovány jako útoky. Běžné detekční systémy provádějí sledování poměru příchozího a odchozího UDP provozu vůči celkovém objemu dat. Bohužel však tyto metriky generují v praxi mnoho falešných poplachů. Pro zvýšení úspěšnosti detekce UDP útoku je vhodné sledovat i ICMP proudy, protože ve většině případů je při UDP útoku generován reverzní ICMP proud. Bohužel však stále zůstává jako jediná možnost detekce sledování provozu a vhodně nastavené hranice UDP provozu.

Případné zmírnění UDP útoků je možné provést omezením rychlosti ICMP odpovědí. Nicméně, takovéto omezení bude mít vliv na legitimní provoz. Tradiční způsob zmírnění UDP útoků také spoléhá na firewally, které filtrují nebo blokují škodlivé UDP pakety. Přesto však tyto metody nejsou dostatečné, pokud bude útok veden s velkým důrazem a intenzitou.[19]

1.3 Zneužití TCP protokolu

TCP je společně s UDP základním protokolem transportní vrstvy. Jedná se o spojově orientovaný protokol se spolehlivým doručováním. Pracuje nad IP vrstvou, která zajišťuje pouze přenos datagramů. Spolehlivost doručování zajišťuje TCP opakovaným odesíláním nespolehlivých či nepotvrzených paketů a přeuspořádáním přijatých paketů do správného pořadí.

Obr. 1.3: Hlavička TCP protokolu.

V rámci TCP se musí před odesíláním dat navázat spojení mezi klientem a serverem. Navazování probíhá pomocí trojcestného handshakingu (three-way handshake), během kterého si obě strany dohodnou číslo sekvence a potvrzovací číslo. Pro navázání spojení slouží datagramy s nastavenými příznaky SYN a ACK.
Průběh navazování TCP spojení:

- Klient odešle datagram s nastaveným příznakem SYN a náhodně vygenerovaným číslem sekvence A, potvrzovací číslo je 0.
- Server odešle datagram s nastavenými příznaky SYN a ACK, potvrzovací číslo je A+1 a náhodně vygenerované číslo sekvence je B.
- Klient odešle datagram ACK, číslo sekvence A+1, číslo odpovědi B+1.

Obě strany si musí pamatovat čísla sekvence své i protistrany, na základě toho je ověřováno korektní doručování datagramů a jejich pořadí. Spojení je navázáno po celou dobu přenosu, což přináší riziko k možnému útoku pomocí SYN flood, který bude popsán dále v kapitole 1.3.2. Pro korektní ukončení spojení se používá podobný postup jako při navazování spojení jen s použitím příznaků FIN a ACK.

- Klient odešle datagram s nastaveným příznakem FIN.

- Server odpoví datagramem s nastaveným příznakem ACK.
- Server odešle datagram s nastaveným příznakem FIN.
- Klient odpoví s nastaveným příznakem ACK.
- Poté je spojení ukončeno.

1.3.1 Zneužití TCP příznaků

Některé typy útoků v rámci testování sítě a zabezpečení systémů mohou rozesílat TCP datagramy s neplatným nastavením příznaků v hlavičce. Na základě zpětných odpovědí dokáže útočník odhalit typ operačního systému, případně identifikaci zařízení pro případné další útoky. [5]

- Žádný TCP příznak: paket, který neobsahuje žádný příznak, není ani inicializací spojení, ani prostředek proudu, ani ukončení nebo reset relace. Paket, který nenese žádý příznak, nemůže být součástí žádné platné TCP transakce.
- SYN/FIN: neplatná kombinace příznaků, která znamená zároveň inicialiazaci a ukončení spojení.
- SYN/RST: neplatná kombinace příznaků, sestavená z kombinace navázaní spojení a resetu spojení.
- SYN/FIN/ACK: kombinace navázaní spojení, potvrzení příjmu a ukončení spojení je další nesmyslná podmínka.
- SYN/RST/ACK: stejně jako předcházející, obsahuje příznak pro navázaní spojení, potvrzení přijmu a pouze místo ukončení spojení obsahuje příznak pro reset spojení.
- Všechny TCP příznaky: tato kombinace všech příznaků se obvykle nazývá vánoční stromeček, celkově se jedná o nesmyslnou kombinaci všech příznaků.

1.3.2 Záplava SYN – SYN flood

Útoky pomocí záplavy SYN využívají chyby v tom, jak je u většiny systémů implementován trojcestný handshake. Když hostitel obdrží žádost SYN od jiného hostitele, odpoví paketem SYN/ACK, vytvoří vstup ve frontě neobsloužených událostí a čeká na příchozí ACK paket pro dokončení trojcestného handshaku. Informace o tomto očekávaném příchozím spojení ukládá do mezipaměti polootevřených oken. Typická časová prodleva (timeout) pro takováto připojení je 3 minuty. Než tato časová prodleva vyprší, může útočník zaslat takovýchto požadavků ohromné množství, čímž dosáhne vyčerpání paměťových prostředků cílového hostitele a jeho vyřazení ze služby.[11]
Útočník může tento typ útoku vést několika možnými způsoby:

- Vysílá záplavu SYN paketů přímo ze své stanice, dokud nedojde k vyčerpání prostředků oběti.
- Může pomocí IP spoofingu podvrhnout zdrojovou IP, aby nebylo možné útočníka detekovat, a odesílá záplavu SYN paketů.
- Útočník pomocí IP spoofingu provede zfalšování zdrojové IP adresy, která se nastaví na adresu oběti. Na vybrané stanice začne posílat SYN pakety a zdrojovou IP adresou nastavenou na IP adresu oběti. Příjemci si myslí, že s nimi chce oběť navázat komunikaci, a odpoví mu paketem SYN/ACK. Oběť nic takového nečeká, standardně by odpověděla paketem RST pro ukončení spojení, ale toho se běžně nevyužívá, a neodpoví tedy nic. Zneužití prostředníci, kteří odesílali SYN/ACK pakety, si vzhledem k časové prodlevě myslí, že se někde ztratily a tak je pošlou znovu. Tím je tento útok ještě zesílen, obvykle se znovuzasílání SYN/ACK opakuje 4x.

1.3.3 Únos TCP spojení

Jedná se o pokračování výše uvedeného Non-Blind Spoofing založeného na podvržení IP adresy. Předpokladem je, že útočník dokáže sledovat spojení mezi dvěma komunikujícími stranami. Poté útočník začne odesílat pakety k cílové stanici s podvrženou IP adresou důvěryhodného zdroje, který komunikuje s cílovým systémem. Odesílané pakety musí mít sekvenční čísla, která cílový systém očekává. Zároveň musí dorazit před pakety z důvěryhodného zdrojového systému. K dosažení tohoto cíle je často nutné důvěryhodný zdrojový systém nějakým druhem DoS útoku vyřadit z provozu, případně pomocí ARP provést únos spojení.

Poté, co bylo TCP spojení převzeto, je již možné komunikovat přímo s cílovým hostitelem s identitou zcizené IP adresy.

1.3.4 Resetovací útoky na TCP spojení

Tyto útoky fungují na podobném principu jako únos TCP spojení. Je třeba znát detaily ohledně komunikace dvou důvěryhodných stran, tedy jejich IP adresy, čísla portů a sekvenční čísla. Útočník následně provede odeslání paketu s podvrženou IP adresou důvěryhodného zdroje a očekávaným sekvenčním číslem. V podvrženém paketu pouze nastaví příznak RST, čímž si cílový hostitel bude myslet, že odesílací strana ukončila spojení a provede uzavření kanálu.[1]

1.3.5 Detekce a prevence zneužití TCP

Pro detekci útoků SYN záplav se používá relativně jednoduchý speciální systém FDS (Flooding Detection System – systém pro detekci záplavových útoků). Jednoduchost tohoto systému spočívá v jeho bezstavovém provozu a nízké výpočetní náročnosti. FDS je v jistém smyslu vedlejším produktem infrastruktury směrovačů, která odlišuje kontrolní TCP pakety od datových paketů. Místo toho, aby sledování provozu probíhalo na vstupní části sítě, jako je firewall či proxy server nebo na samotných cílových stanicích, probíhá detekce SYN záplav na směrovačích, které připojují koncové stanice do internetu. FDS lze nasadit na směrovačích v první míli nebo v poslední míli sítě. Výhodou nasazení FDS v první míli je jejich blízkost ke zdroji útoku. Pokud bude zjištěn útok SYN záplavou na router v první míli, tak je zároveň zachycen i zdroj útoku, protože zdroje zaplavení musí být uvnitř podsítě, na který je daný router připojen a lze jej tedy zpětně vysledovat.[17]

Základním obranným prvkem proti záplavám SYN jsou SYN cookies. Útok záplavou SYN je založen na zahlcení oběti zprávami SYN a tím dosažení jejího vyčerpání zdrojů a následnou nedostupností. Pokud se použijí SYN cookies, tak server po obdržení žádosti o navázání spojení pošle klientovi odpověď a danou žádost vyhodí z fronty polootevřených spojení. Do odpovědi, kterou odesílá, nevkládá náhodné číslo sekvence, ale číslo, které je vygenerované podle daných pravidel. Když server od klienta obdrží potvrzení odpovědi, tak je podle daného čísla sekvence schopen zpětně odvodit, zda se jedná o korektní paket. Pokud sekvenční číslo souhlasí, je spojení navázané.[2]

Resetovacím útokům na TCP spojení lze předcházet pomocí detekčních nástrojů, kde je vhodně nastavena hranice maximálního možného výskytu RST paketů. Danou hranici je vhodné nastavit až po několikadenním sledování provozu a následném vyhodnocení výskytu RST paketů.

1.4 Zneužití ICMP protokolu

ICMP je jedním z nejdůležitějších síťových protokolů. Je používán síťovými zařízeními, nejčastěji routery, pro zasílání chybových zpráv, které například uvádějí, že požadovaná služba není dostupná nebo že není možné dosáhnout cílového hostitele nebo router. ICMP může být také použit pro přenos zpráv dotazu. Má přiřazeno číslo portu 1. ICMP se liší od transportních protokolů jako jsou TCP a UDP v tom, že není obvykle používán pro výměnu dat mezi systémy, ani není pravidelně využíván koncovým uživatelem síťových aplikací, s výjimkou některých diagnostických

nástrojů, jako je ping a traceroute.
Nejpoužívanější ICMP zprávy:

- Echo request – požadavek na odpověď, na kterou by měli zareagovat všechny prvky na IP vrstvě.
- Echo reply – odpověď na požadavek.
- Destination unreachable – zpráva o nedostupnosti, která je ještě upřesněna, zda se jedná nedostupnou síť (Net unreachable), nedostupnou stanici (Host unreachable), nemožnost využití protokolu (Protocol unreachable) nebo nedostupnosti konkrétního portu (Port unreachable).
- Redirect – zpráva o přesměrování provozu, používá se, pokud k cíli vede lepší cesta.
- Time to Live exceed in Transit – vypršení časového limitu.

1.4.1 Záplava ICMP – ICMP flood

K zaplavení pomocí ICMP obvykle dochází, když útočník zasílá oběti velké množství zpráv ICMP echo request. Jedná se o zprávy, na které musí oběť reagovat, proto musí vynaložit všechny své zdroje k odpovědi. Aby byl útok efektivní, je třeba, aby měl útočník vyšší kapacitu připojení a oběť žádnou nebo minimální ochranu. Tím může dojít k vyčerpání kapacity připojení oběti a jejímu odpojení od sítě.

1.4.2 Smurf útok

Smurf je další druh DoS útoku. Je podobný ICMP záplavě, ale zprávy ICMP Echo request nejsou zasílány na konkrétní stanice, ale na adresu sítě. To způsobí, že všechny stanice v síti vrací reakci ICMP Echo reply. V závislosti na rozloze konkrétní sítě to vytváří vysoký síťový provoz a síť je zahlcená takovým způsobem, že přestane odpovídat.[21]

1.4.3 Ping of Death

Jedná sice o starý a již překonaný útok, ale ve své době byl velmi významný a je třeba jej zmínit. Jedná se o útok, který zneužíval chyby v implementaci IP a ICMP protokolů v operačních systémech. Tento útok je postaven na paketu ICMP Echo request. Dle specifikace ICMP je maximální délka ICMP Echo paketů maximálně 65535 bajtů. Toho bylo možné dosáhnout zasíláním fragmentovaného IP paketu, který obsahoval ICMP Echo request a po složení přesáhl velikost 65535 bajtů. Některé operační systémy na to nebyly připraveny a při obdržení této zprávy kolabovaly. Postupnou aktualizací a vydáváním hotfixů byla tato chyba byla odstraněna a lze říci, že tento útok již je mrtvý.

1.4.4 Detekce a prevence zneužití ICMP

Detekce útoků pomocí ICMP zpráv není příliš jednoduchá. Je třeba rozlišit, zda je útok veden z jedné stanice nebo z botnetu. Pokud je útok vedený z jedné stanice, stačí se zaměřit na množství obdržených ICMP od jednoho odesílatele a je jedno, zda je IP adresa platná či podvržená. Naopak detekce ICMP útoku, vedená pomocí sítě ovládnutých počítačů, již pro detekci tak snadná není. ICMP zprávy jsou vyžadovány pro komunikaci síťových zařízení. ICMP je protokol, který má být použit pro upozornění hostitele o problémových podmínkách nebo pro výměnu zpráv. Nicméně použití zpráv škodlivým způsobem umožňuje útočníkovi získat informace o hostiteli a topologii sítě. Nastavit systém detekce pro aktivní monitorování sítě na detekci podvržených ICMP je pracné. Musí být provedeno vhodné filtrování ICMP, aby se minimalizovala potenciální hrozba.

Při zapnutí funkce ochrany před záplavou ICMP zpráv lze nastavit práh, který při překročení vyvolá funkci ochrany před povodněmi ICMP. Pokud je prahová hodnota překročena, bezpečnostní zařízení ignoruje další zprávy typu ICMP Echo Request po zbytek stávající vteřiny plus příští vteřinu. Pro další omezení útoku je vhodná analýza systému a ujasnit si jaké ICMP zprávy mohou být očekávány. V případě, že nějaký typ zpráv není cílovým systémem podporován, je vhodné jej zablokovat. Pro detekci ICMP útoků se nabízejí speciální IDS, kde je možné tyto parametry a vlastnosti upravovat.[15]

Jako prevence před útoky typu Smurf je nejdůležitější zakázat broadcastové adresování na vstupních routerech a firewallech. Většina starších síťových zařízení má ve výchozím nastavení povoleno adresování na broadcastu. Proto je důležité, aby na všech zařízeních a rozhraních, kde to není bezpodmínečně vyžadováno, byla broadcastová adresace vypnutá.[10]

Poslední z ICMP útoků byl zmíněn Ping smrti. Tento druh útoku v současnosti již nehrozí. Útok byl založen na chybách v implementaci IP a ICMP v rámci operačních systémů. Výrobci systémů tyto chyby opravili zhruba v roce 1997 a aktualizovaný operační systém by s tímto útokem neměl mít sebemenší potíže.

1.5 Zneužití DNS

DNS – Domain Name System je internetová služba, která zajišťuje překládání názvů domén na IP adresy. Vzhledem k tomu, že doménová jména jsou složená z písmen a slov, jsou snadněji zapamatovatelné než číselné IP adresy, na jejichž

principu je směrování v internetu založeno. Pokaždé, když se použije název domény, musí DNS provést přeložení názvu na odpovídající IP adresu.

1.5.1 DNS zesilující útok

Zesílený DNS útok je založený na útoku typu DDoS. Útočník podvrhuje dotazy na DNS s tím, že skryje svou identitu a zároveň řídí odpověď k cíli. Prostřednictvím různých technik se DNS dotaz může změnit v ohromné zatížení cílové sítě. Útočník pošle DNS dotaz s podvrženou cílovou IP adresou na DNS, která jsou zranitelná. Nejčastěji se jedná o open DNS servery. Původní požadavek je často přenášen přes botnet pro větší základny útoku a další utajování. Pro vytvoření velkého DNS dotazu lze použít rozšíření EDNS. Do zpráv je pro další zvětšení jejich velikosti možné přidat rozšířené zabezpečení – DNSSEC. Na základě těchto rozšíření se zvětší velikost paketu na 4000 bajtů, což je víc než maximální velikost ethernetového paketu a tím pádem musí být při přenosu fragmentován. To způsobuje další navýšení síťových zdrojů a zahlcení sítě.[13]

1.5.2 Detekce a prevence zneužití DNS

Cílem útoku na DNS je posílat velké objemy síťového provozu na DNS server, aby byl zahlcen a přestal komunikovat vůči ostatním účastníkům. Běžným příznakem bývá, že server přestane reagovat po síti. Na přímé připojení bude reagovat bez potíží. Při podezření na probíhající útok lze použít program tcpdump na prověření zda DNS server není zahlcen velkým množstvím odpovědí, o které nežádal.[13]

Jako prevence před útoky na DNS je doporučováno upravit zabezpečení samotného DNS serveru, protože největší část útoků je směrována na open DNS.

1.6 Ostatní útoky

Ostatní útoky si už zmíníme pouze krátce. Povětšinou se jedná o útoky, které jsou vedeny jako individuální útoky na konkrétní protokoly, nejčastěji aplikační vrstvy nebo zneužívají chyb v aplikacích či systémech. Nelze je tedy brát jako hromadné útoky a tedy jejich provedení by nebylo možné v rámci detekce anomálií datového provozu odhalit.

1.6.1 Přetečení zásobníku – Buffer overflow

K přetečení zásobníku dojde, když se program nebo proces pokusí ukládat více dat do vyrovnávací paměti, než bylo původně zamýšleno. Vzhledem k tomu, že vy-

rovnávací paměti jsou vytvořeny tak, aby obsahovaly pouze omezené množství dat, tak další informace, které by měly být zapsány, mohou přetéct do přilehlých vyrovnávacích pamětí a tím způsobit zničení nebo přepsání platných dat v nich zapsaných. Přetečení může nastat z důvodu programové chyby, ale stále častěji se jedná o bezpečností útok na integritu dat. Při těchto útocích mohou data, která přetečou zásobník, obsahovat kódy, jejichž cílem je vyvolat specifické akce. Mohou obsahovat pokyny pro napadené počítače, které by mohly například poškodit soubory uživatele, změnit údaje nebo sdělit důvěrné informace.[18]

1.6.2 Zadní vrátka – Backdoor

Zadní vrátka jsou druhem přístupu k počítačovému programu, který obchází bezpečnostní mechanismy. Programátor může někdy vytvořit zadní vrátka, aby byl k programu umožněn přístup i jiným způsobem, např. z důvodů řešení problémů. Nicméně tato zadní vrátka bývají zneužívána i útočníky k neoprávněnému přístupu do systému. Pro tyto účely nasazují útočníci škodlivé programy – červy, kteří tato vrátka vyhledávají a otevírají cestu případným útočníkům.

2 TEORETICKÁ ANALÝZA ÚTOKŮ

Nyní, když byly popsány nejčastější a nejvýznamnější datové útoky, je třeba provést teoretickou analýzu a vybrat útoky, jejichž průběh bývá natolik výrazný, že by mohly být detekovány jako anomálie datového provozu. Postupně si zrekapitulujeme výše uvedené záplavové útoky a pokusíme se rozhodnout, zda je možné, aby se jejich průběh projevil jako datová anomálie. Stále je třeba brát v úvahu, že anomálie v datovém provozu mohou způsobit pouze útoky nejsilnějšího rázu, tedy hlavně záplavové útoky. Ostatní typy útoků pravděpodobně v datovém provozu žádnou anomálii nezpůsobí a nebylo by je tedy možné detekovat.

2.1 IP útoky

Při popisu zneužití protokolu IP byly zmíněny útoky IP záplavou. Tento útok je založen na zaplavení oběti z různých podvržených IP, proto v rámci analýzy datového provozu je třeba sledovat výrazné nárůsty paketů na stejnou IP adresu.

2.2 UDP útoky

Detekci UDP záplavových útoků lze provádět pouze na základě porovnání normálních hodnot UDP provozu vůči naměřeným hodnotám. Proto bude vhodné provést velké množství nezávislých měření pro stanovení průměrného UDP provozu, též bude třeba stanovit odchylku, aby bylo možné odhadnout, zda případná anomálie na UDP provozu může či nemůže být útok.

Zároveň při útoku záplavou UDP by měl být detekovatelný proud zpětných zpráv ICMP. Takže pro lepší detekci je vhodné provést pravidlo pro kontrolu příchozího a odchozího provozu zároveň, což zvýší pravděpodobnost úspěchu detekce útoku.

2.3 TCP útoky

Detekce anomálií, které souvisejí s TCP záplavovým útokem, jsou asi nejsložitější částí. Důvodem je celková kombinace složitosti TCP hlavičky i samotný průběh útoku. Pro samotnou detekci na úrovni datového provozu existuje několik specifických algoritmů, které jsou zaměřeny na porovnávání počtu paketů s příznaky SYN a FIN, případně RST v určitém časovém intervalu. Detailní popis samotných algoritmů je vysvětlen například v dokumentu z Michiganské univerzity[17].

V běžném datovém provozu lze sledovat závislost mezi množstvím SYN a FIN nebo RST paketů, počet SYN, FIN a RST paketů by měl korelovat. Proto je vhodné provádět sledování výskytu jednotlivých paketů během časového intervalu, např. 10 minut, a porovnávat poměry. Vyšší výskyt určitého druhu by mohl znamenat útok. V případě, že výrazně vzroste množství SYN paketů, by se mohlo jednat o záplavu SYN, pokud bude vyšší výskyt FIN nebo RST paketů, tak by se mohlo jednat o resetovací útok.

2.4 ICMP útoky

Detekci ICMP útoků lze realizovat na podobném principu jako UDP. Je třeba sledovat průměrné hodnoty ICMP provozu a porovnávat s okamžitou hladinou.

2.5 DNS útoky

Protože útoky DNS útoky jsou směrovány na jednotlivé DNS servery, není v rámci detekce provozu možné útok rozeznat podle zdrojových a cílových IP adres rostoucího DNS provozu. Je tedy možné pouze sledovat množství výskytu DNS dotazů v provozu a při nárůstu je možné maximálně konstatovat, že někde v síti by mohl probíhat útok na DNS.

3 STATISTICKÉ METODY

V této práci budou pomocí různých statistických metod analyzována data získaná z reálného provozu.

3.1 Časové řady

Protože by měl být provoz zaznamenáván v pravidelných intervalech po určité časové období, mělo by být možné data analyzovat pomocí časových řad.

Pro základní práci s časovými řadami poslouží nejlépe grafy. Dále je vhodné počítat se základními statistickými hodnotami:

- prostého aritmetického průměru – nejznámější odhad střední hodnoty

$$\overline{y} = \frac{\sum_{i=1}^{n} y_i}{n} \tag{3.1}$$

- rozptylu – druhý centrální moment, který bývá definován jako střední hodnota kvadrátů odchylek od střední hodnoty

$$s_y^2 = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \overline{y}) \tag{3.2}$$

- směrodatné odchylky – odmocnina rozptylu

$$s_y = \sqrt{s_y^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (y_i - \overline{y})} \tag{3.3}$$

Pomocí dalších statistických výpočtů by bylo možné získat míry dynamiky, které umožňují charakterizovat detailní rysy chování, jako jsou přírůstky a koeficienty růstu a pro porovnání více vzorků mezi sebou počítat korelace provozu.[23] Tyto metody by bylo vhodné použít při dlouhodobém sledování provozu a nejsou v této práci použity. Při dlouhodobém sledování provozu by také bylo vhodné sledovat trendy jednotlivých časových řad. V poslední době dochází k neustálému narůstání datovému provozu a při dlouhodobém sledování by toto mělo být znatelné.

3.2 Popisná statistika

Druhou možností, jak budou získaná data statisticky analyzována, je použití popisné statistiky. Na získané vzorky lze pohlížet jako na diskrétní data a poté nestačí provozy charakterizovat jenom střední hodnotou. Kromě aritmetického průměru,

rozptylu a směrodatné odchylky se pro bližší prozkoumání provozu využívá také kvantilových hodnot. Jedná se o hodnoty, které dělí uspořádaný statistický provoz na určitý počet stejně obsazených částí. Pro potřeby této práce jsou využity hodnoty dolního, horního a středního kvartilu (medián). Medián představuje střední hodnotu statistického souboru vzestupně seřazených hodnot. S doplněním maximální a minimální hodnoty z daného souboru je možné vytvořit krabičkový graf, který následně podává informaci o poloze, variabilitě a rozložení hodnot daného vzorku.

Dále je možné a vhodné určovat variační koeficient datového vzorku:

$$V = \frac{s_y}{\overline{y}} \cdot 100[\%] \tag{3.4}$$

Pokud je hodnota variačního koeficientu větší než 50 %, tak získaný vzorek vykazuje vysokou nesourodost a další výpočty založené na těchto hodnotách by byly zkresleny.

Vždy, pokud má být nějaký datový soubor statisticky analyzován, je vhodné určit jeho rozložení. Nejvhodnější je, pokud má vzorek normální rozložení. S takovýmto souborem dat lze přímo pracovat a není třeba jej nijak upravovat. Pro určení normality statistického souboru existuje velké množství postupů. Základní ověření, zda daný datový soubor může odpovídat normálnímu rozdělení lze provést pomocí koeficientů šikmosti a špičatosti.

Po vypočtení a vyhodnocení koeficientu šikmosti lze předpokládat, zda se rozdělení dat podobá normální křivce. Jeho výpočet vychází z třetího centrálního momentu:

$$\alpha = \frac{\sum_{i=1}^{n} (y_i - \overline{y})^3}{n s_y^3} \tag{3.5}$$

Koeficient šikmosti může nabývat hodnot:

- $\alpha = 0$ – přibližně symetrické rozdělení
- $\alpha > 0$ – rozdělení zešikmené doleva
- $\alpha < 0$ – rozdělení zešikmené doprava

Koeficient špičatosti vychází ze čtvrtého momentu směrodatné proměnné:

$$\beta = \frac{\sum_{i=1}^{n} (y_i - \overline{y})^4}{n s_y^4} - 3 \tag{3.6}$$

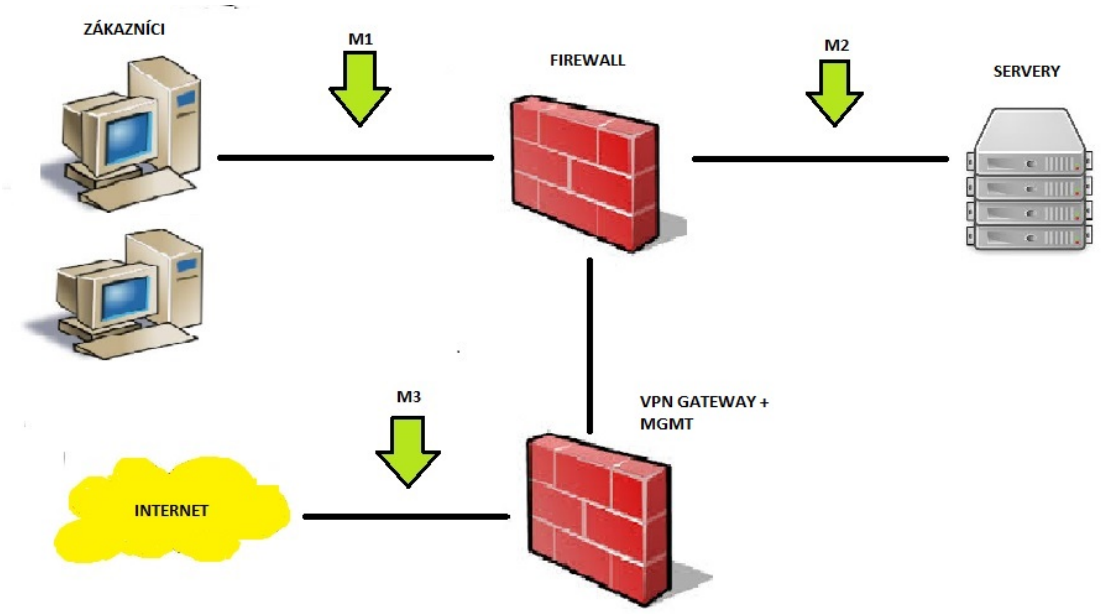
U normálního rozdělení je tento moment roven 3 a pak tedy koeficient špičatosti je roven nule. Je-li koeficient špičatosti větší než nula, tak je toto rozdělení špičatější než normované normální rozdělení, pokud je menší než nula, tak je dané rozdělení plošší.

Hrubý odhad normality dat lze také získat porovnáním aritmetického průměru a mediánu. Tyto hodnoty by se neměly lišit o více jak 10 %. Jako další možnost pro ověření normality zdrojových dat je vytvoření histogramu četnosti vzorků a P-P grafu, který zobrazuje vztah hodnot kumulativní distribuce hypotetického normálního rozdělení a hodnot kumulativní distribuce zkoumaného souboru. Pokud bude sledovaný vzorek mít normální rozdělení, budou body ležet na přímce.[24]

4 VYUŽITÍ STATISTIKY K DETEKCI PROVOZNÍCH ANOMÁLIÍ

4.1 Příprava souborů k analýze

Záznam datového provozu byl prováděn na 3 sondách v období od 23.3.2015 12:00 do 13.4.2015 3:00. Na obrázku 4.1 je zobrazeno rozmístění jednotlivých záznamových sond v rámci infrastruktury Cesnetu.



Obr. 4.1: Umístění jednotlivých záznamových sond v infrastruktuře Cesnetu.

Vzhledem k velkému objemu datového provozu v infrastruktuře Cesnetu bylo odchytáváno pouze prvních 64 bytů přenášených paketů, aby bylo možné analyzovat hlavičky jednotlivých paketů a získat tak data potřebná k analýzám. Protože se jednalo o veřejný provoz, byl z důvodu ochrany osobních údajů obsah anonymizován. Záznam byl automaticky ukládán v hodinových úsecích. Během pořizování záznamu bohužel došlo k výpadku měření v období mezi 1. a 4.4., takže záznam není nepřetržitý. I tak se celkově jednalo o data v objemu téměř 360 GB[1].

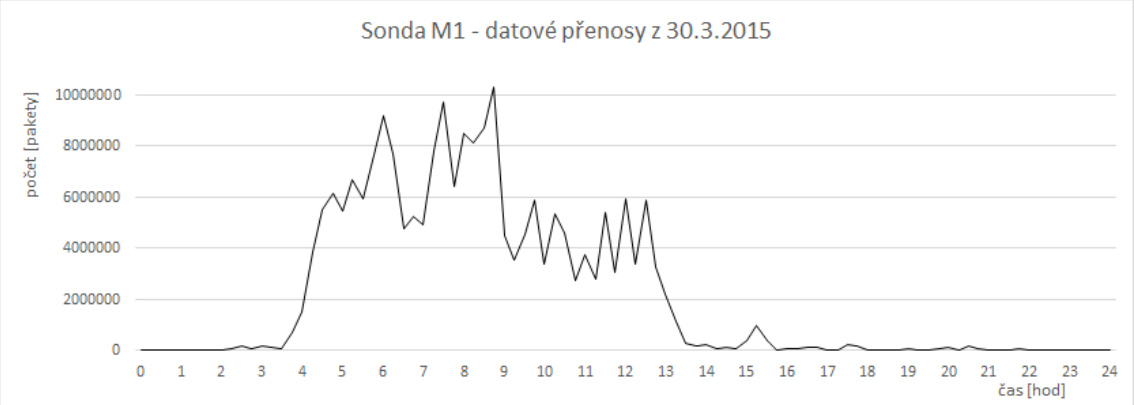
Zdrojové soubory dodané pro analýzu provozu byly ve formátu *pcap*. Pro další analýzu je tento typ souborů plně vyhovující a nebylo tedy nutné provádět převod

[1]Data jsou dočasně uložená na cloudovém úložišti Cesnetu.

na jiný typ souborů. Při pořizování záznamu bylo dopředu počítáno s tím, že budou soubory dále upravovány a proto byly všechny záznamy zahajovány a ukončovány vždy v celou hodinu. Pokud by bylo třeba tyto časy upravit, lze použít např. program editcap, který je součástí balíku aplikace Wireshark[2]. Pro případnou úpravu lze využít následující syntaxi. Dle potřeby konkrétního záznamu se nastaví vhodné datum a čas začátku a konce záznamu, Dále je třeba nahradit platný název vstupního souboru a zvolit jméno výstupního souboru.

```
editcap -A '2015-03-30 00:00:00' -B '2015-03-30 01:00:00'
    vstupni.pcap vystupni.pcap
```

Celkově tedy zdrojová data obsahovala téměř 500 hodinových záznamů provozu pro každou sondu. Protože s takovýmto objemem dat nelze jednoduše pracovat, bylo třeba zvolit konkrétní hodinu, jejíž provoz by byl dále analyzován. Z celého souboru byl náhodně zvolen jeden pracovní den, který posloužil jako náhled na průměrný denní provoz. Podle velikostí souborů a tedy i objemů provozu bylo zvoleno úterý 30.3.2015. 24 hodinových souborů z celodenního provozu bylo rozděleno na 15minutové úseky. Všechny soubory s těmito 15minutovými záznamy byly uloženy do společného adresáře a pomocí skriptu *parse_den.ksh*, viz příloha A.1, byly spočteny celkové počty paketů v jednotlivých časových úsecích. Získané hodnoty byly přeneseny do grafu, který souhrnně zobrazuje denní provoz po 15minutových intervalech.



Obr. 4.2: Graf datových přenosů na sondě M1 za 24 hodin.

Z grafu 4.2 lze vyčíst, že provozní špička se v daném dni nacházela mezi 8:00 a 9:00 a tato hodina byla zvolena pro detailnější analýzu. Tím se vstupní soubor zmenšil na sadu hodinových záznamů pro každý den, kdy záznam běžel.

[2]Kompletní aplikace je volně ke stažení na adrese: https://www.wireshark.org/

Pro větší detaily provozu byly hodinové intervaly dále rozděleny na minutové úseky. Rozdělení bylo provedeno pomocí příkazu:

```
editcap -i 60 vstupni.pcap vystupni.pcap
```

Číslice 60 v příkazu značí, že soubory jsou děleny po 60 vteřinách. V případě potřeby jiného časového intervalu lze toto číslo změnit na požadovanou hodnotu.

Z každého takto vytvořeného souboru minutových vzorků byl vybrán vždy pátý soubor pro samotné vypočítání konkrétních sledovaných atributů datového přenosu. Takto se podařilo zdrojová data zmenšit na necelých 55 GB a vzhledem k velikosti byla uložena na webovém úložišti One Drive[3].

4.2 Získání dat

Ze vstupního souboru bylo třeba získat data, která by mohla být statisticky hodnotná a zároveň charakterizující datový provoz. Zároveň bylo třeba získat hodnoty charakteristik, jejichž výkyv by mohl pomoci detekovat určitý druh útoku. Pro hlavní analýzu byly vytvořeny 4 skripty pojmenované dle sledovaných vlastností. Programová logika všech skriptů je totožná. Po spuštění skripty provedou nastavenou analýzu pro všechny *pcap* soubory v daném adresáři. Na základě teoretické analýzy datových útoků (kapitola 2) byly jako sledované charakteristiky zvoleny velikosti přenášených paketů, TCP příznaky a komunikace pomocí konkrétních portů a protokolů. Data získaná během skriptů jsou zaznamenávána do výstupního souboru typu *csv*. Všechny skripty které byly pro potřeby této práce vytvořeny, jsou uvedeny v příloze a zároveň uloženy v elektronické příloze. Skripty byly vytvořeny v bashi pro jejich přehlednost a snadné nasazení. Zároveň musely být co nejjednodušší z důvodu velkých objemů dat, které musely zpracovat, zanalyzovat a zaznamenat na výstup.

Kostra skriptů je ve všech případech zhruba stejná. Skript po spuštění vytvoří *csv* soubor pro výstupní data. V případě opakovaného spuštění stejného skriptu ve stejném adresáři přepíše původní data. Poté jsou definovány proměnné pojmenované logicky dle sledovaných faktorů. Následuje cyklus, který prochází postupně všechny *pcap* soubory ve složce, kde se skript nachází, a sekvence příkazů programu tshark, který byl zvolen pro analýzu datového provozu. Program tshark je terminálová aplikace, která provádí analýzu aktuálně načteného souboru s pomocí filtrů, které fungují i v aplikaci Wireshark a získaná data je tak možné ověřit i v grafickém prostředí[22]. Stejně jako výše uvedená aplikce editcap je tshark součástí instalačního balíku Wireshark.

[3]Data jsou volně přístupná na adrese http://1drv.ms/1R7jY6n

V následující ukázce části skriptu *flags.ksh* (kompletní skript v příloze A.3) je cyklus, který pro všechny *pcap* soubory v adresáři, provede sekvenci tshark příkazů. S jejich pomocí jsou sečteny různé výskyty TCP příznaků a tyto počty jsou pro každý analyzovaný *pcap* soubor zaznamenány na nový řádek výstupního *csv* souboru včetně názvu analyzovaného souboru. Když jsou takto zpracovány všechny soubory v adresáři, tak jsou vyčištěny dočasné proměnné a cyklus je ukončen.

```
for PCAPFILE in `ls *.pcap`; do
    echo "doing: $PCAPFILE"
    SYN=`tshark -r $PCAPFILE -Y "tcp.flags.syn == 1 &&
        tcp.flags.ack == 0" | wc -l`
    ACK=`tshark -r $PCAPFILE -Y "tcp.flags.ack == 1" | wc -l`
    SYNACK=`tshark -r $PCAPFILE -Y "tcp.flags.syn == 1 &&
        tcp.flags.ack == 1" | wc -l`
    PUSHACK=`tshark -r $PCAPFILE -Y "tcp.flags.push == 1 &&
        tcp.flags.ack == 1" | wc -l`
    RST=`tshark -r $PCAPFILE -Y "tcp.flags.reset == 1" | wc -l`
    FIN=`tshark -r $PCAPFILE -Y "tcp.flags.fin == 1" | wc -l`
    echo "$PCAPFILE;$SYN;$ACK;$SYNACK;$PUSHACK;$RST;$FIN;" >> $RESULTS
    SYN=0; ACK=0; SYNACK=0; PUSHACK=0; RST=0; FIN=0
done
```

4.3 Popis skriptů

Data byla získávána pomocí 4 skriptů napsaných v bashi. Každý z nich zaznamenával hodnoty jiných atributů provozu. Snahou bylo, aby skripty získaly z analyzovaných souborů taková data, která by při následné analýze mohla napomoci k případné detekci útoku.

Skript *parse.ksh* (viz příloha A.2) sčítá v každém sledovaném souboru výskyty paketů ve velikostech z intervalů:

- 0–300 kB,
- 301–600 kB,
- 601–900 kB,
- 901–1200 kB,
- 1201–1500 kB
- a více jak 1500 kB.

V průběhu řešení bylo třeba řešit úpravu skriptu. V testovací verzi bylo v tshark syntaxi použito parametru `packet.cap_len`, který odečítá velikost právě načteného paketu v rámci analyzovaného souboru. Protože při záznamu na Cesnetu byla velikost odchytávaných paketů zmenšena na 64 kB, tak při první analýze dodaného provozu vycházelo, že velikost všech paketů je v intervalu 0–300 kB. Proto byl parametr v tshark syntaxi upraven na `packet.len`, který odečítá délku paketů z číselného údaje v hlavičce paketu a následně již byla získána korektní data.

Druhý skript *flags.ksh* byl zobrazen výše a sleduje TPC příznaky v hlavičkách všech paketů a zaznamenává výskyty příznaků:

- Pouze SYN,
- ACK v libovolné kombinaci,
- Kombinace SYN/ACK,
- Kombinace PSH/ACK,
- RST,
- FIN.

Třetí a čtvrtý skript spolu úzce souvisí. Skript *protocol.ksh* (příloha A.5) sleduje výskyty paketů TCP, UDP a ICMP. Na něj navazuje skript *ports.ksh* (příloha A.4), který počítá výskyty paketů přenášených na specifické porty aplikačních protokolů s tím, že byly vybrány ty nejobvyklejší, např. HTTP, HTTPS, DNS a další. V tomto skriptu se pak ještě počítaly souhrnné výskyty paketů TCP a UDP na známých portech 0-1023.

Výpočty byly vzhledem k časové náročnosti každého z nich prováděny v několika zároveň spuštěných terminálových oknech aplikace Cygwin64 Terminal[4] pod systémem Windows 10 TP na notebooku s procesorem Intel Core i7-3537U a 12 GB RAM. Původně bylo využíváno linuxového terminálu v systému Ubuntu 14.04 LS, který byl instalován jako virtuální systém pomocí aplikace VirtualBox. Zde však docházelo k vyčerpávání paměťových prostředků a proto bylo od tohoto postupu upuštěno a výpočty byly spouštěny přímo na hlavním systému.

4.4 Zpracování dat

Výstupem ze všech vykonaných skriptů na všech sondách byly *csv* soubory obsahující název jednotlivých analyzovaných *pcap* souborů a hodnoty všech sledovaný

[4]Volně dostupné na https://www.cygwin.com/

parametrů provozu. Dle konkrétních sond byly *csv* soubory sloučeny do sešitů MS Excel. Tyto soubory byly duplikovány s tím, že jedna verze souborů obsahuje kompletní provoz a ve druhé verzi byly ponechány výsledky pouze za pracovní dny, které byly dále analyzovány.

Ve výstupních *csv* souborech jsou na nejvyšším intervalu zachyceny pakety pouze na sondě M3. Sondy M1 a M2 mají pravděpodobně nastavenu MTU menší než 1500 kB a tedy na tomto intervalu nebyl zaznamenán žádný provoz.

Vzhledem k tomu, že dopředu nebyla známa charakteristika zaznamenaného provozu, byla při kontrole výstupních *csv* souborů většina sledovaných portů odmazána, protože se daný provoz buď vůbec nevyskytoval, nebo byl obsažen pouze v minimální míře, která v poměru s celkovým provozem nemá žádnou hodnotu. Zároveň se výrazně zmenšil a zpřehlednil výstupní datový vzorek.

4.5 Statistická analýza

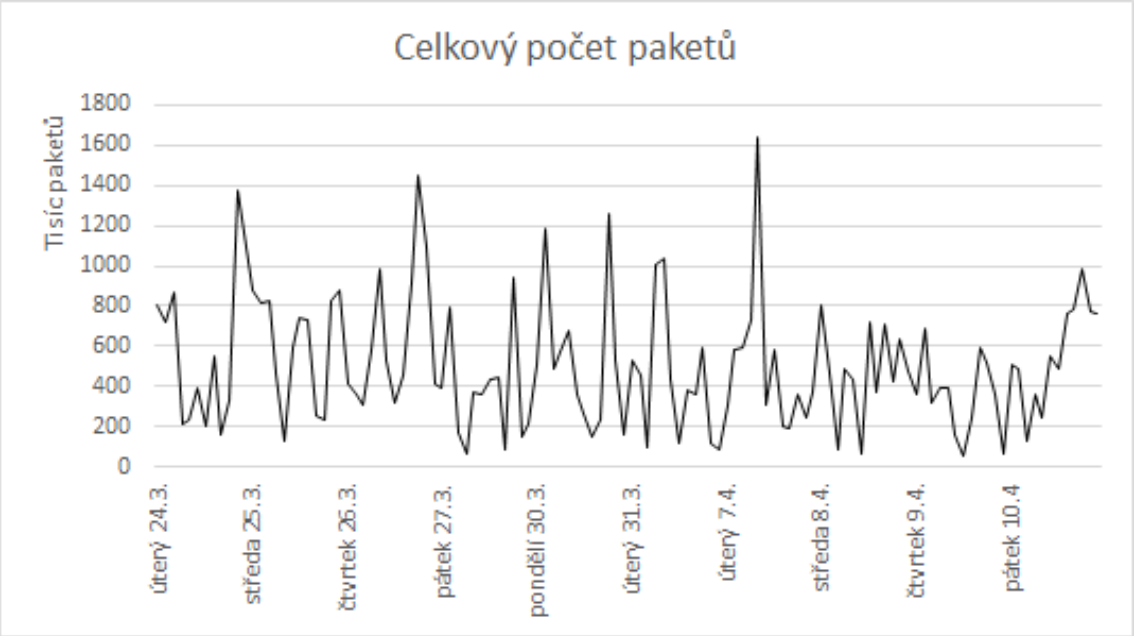
Získaná data jsou nyní připravena ke statistickým analýzám.

4.5.1 Časové řady

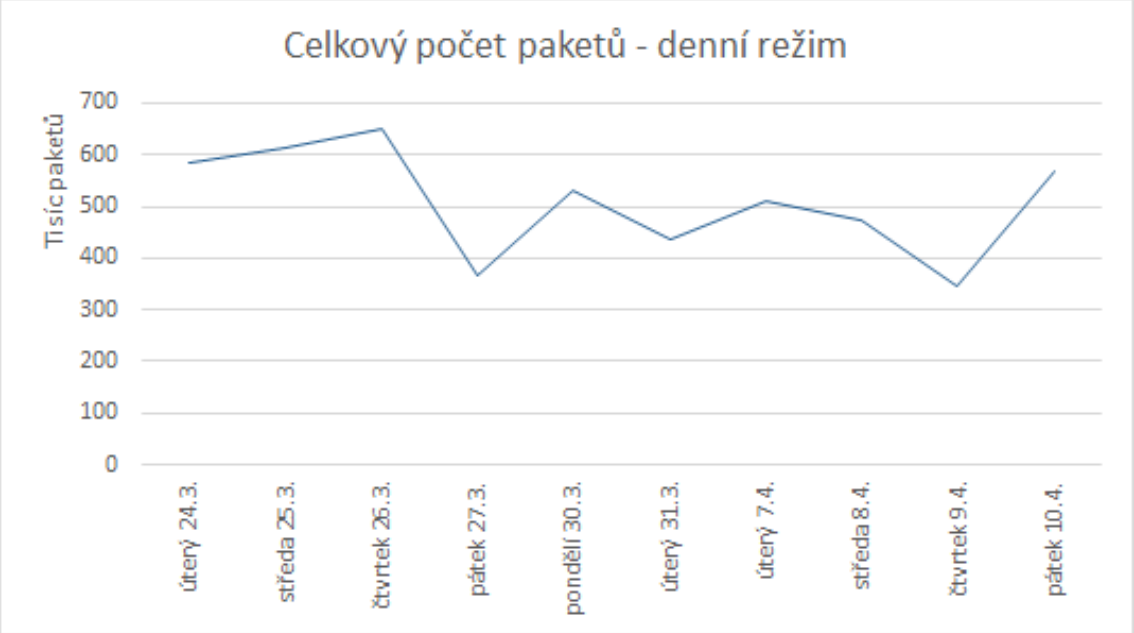
Na zaznamenaná data se nelze dívat jako na správnou časovou řadu. Ideální časová řada bývá definována jako posloupnost hodnost sledovaných parametrů, které jsou zpravidla měřeny v rovnoměrných časových intervalech. Zde se prolínají dva časové intervaly. Krátkodobý záznam, který je představován minutovými intervaly v rámci jedné hodiny (8:00 – 9:00) a dlouhodobý, který představuje opakování stejné hodiny v denním režimu po dobu, kdy záznam probíhal (23.3. – 13.4.). I tak byla získaná data zobrazena do grafů, zda by bylo možné vyčíst nějaké pravidelné chování. Soubory s daty a grafy s analýzou časových řad jsou uloženy v elektronické příloze v adresáři Časové řady.

Celkový provoz na sondě M1 je zobrazen na následujících grafech. Oba grafy zobrazují přenesené pakety na této sondě během stejného časového období. První graf byl sestaven ze všech prvků získaných z minutových *pcap* souborů – obrázek 4.3. V druhém grafu (obrázek 4.4) jsou zprůměrovány předchozí minutové výskyty. Tento graf má sice hladší průběh, ale obsahuje méně vzorků a pro statistické výpočty by bylo třeba dlouhodobé sledování, aby získané hodnoty měly větší vypovídající hodnotu. Zde by bylo vhodné zvážit nasazení databázového systému pro ukládání

výstupů z měření. Pro práci s velkým množstvím získaných vzorků již práce s *csv* soubory a sešity Excelu přestává být přehledná a vrůstá riziko chyb.



Obr. 4.3: Graf znázorňující celkové datové přenosy na sondě M1 v minutových intervalech.



Obr. 4.4: Graf datových přenosů na sondě M1 – denní průměry.

Z grafů lze vyčíst informaci, že při běžném provozu, který byl sledován sondou M1, se přenáší až 1800000 paketů/min ve špičkách a průměrný provoz je mezi 300000 a 700000 pakety/min. Graf denních průměrů provoz celkově vyhlazuje. Při analýzách je vhodné prověřit graf denních průměrů a v případě zjištění odchylky od standardního provozu detailně zanalyzovat v rámci minutového provozu.

Takto jsou zpracovány získané hodnoty u všech sond a všech druhů měření. Veškeré grafy byly prohledány z hlediska vzájemné podobnosti hodnot a možného budoucího využití. Zároveň byly porovnávány charakteristiky průběhů grafů a hledány různé podobnosti mezi průběhy jednotlivých grafů.

V tabulce 4.5 je vidět, že průměrná hodnota celkového počtu paketů v provozu M1 je 507776,2917 paketů/min. Vypočtená hodnota tedy souhlasí s odhadovanou hodnotou získanou z grafu denního provozu. Dále byly vypočteny hodnoty rozptylu a směrodatné odchylky. Případné vysoké hodnoty odchylky a rozptylu mohou naznačit, že analyzované vzorky nemusí být vhodné pro další práci. Podobné chování lze sledovat u velké části ostatních měření. Proto nelze většinu hodnot použít v surovém stavu, ale je potřeba uvažovat souvislosti.

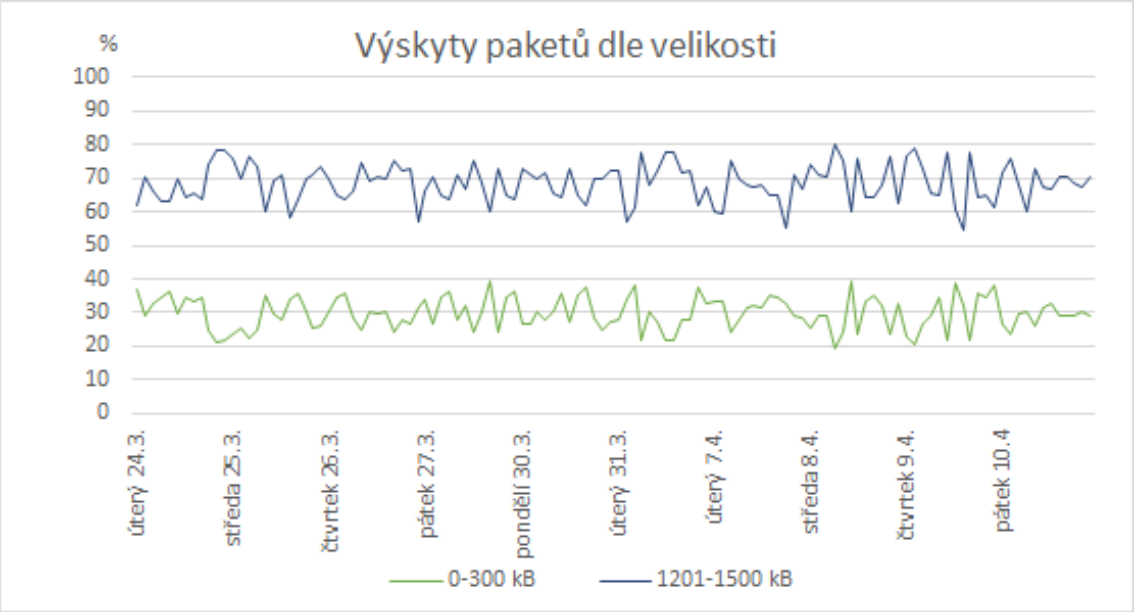
Statistické hodnoty							
	total		0-300 kB	301-600 kB	601-900 kB	901-1200 kB	1201-1500 kB
Minutová měření							
aritmetický průměr	507776,2917		29,8190	0,7209	0,1400	0,6772	68,6430
rozptyl	101355584912,7900		22,6122	1,5222	0,0249	1,4610	31,6129
směrodatná odchylka	318363,919		4,7552	1,2338	0,1577	1,2087	5,6225
Denní průměry							
aritmetický průměr	507776,2917		29,8190	0,7209	0,1400	0,6772	68,6430
rozptyl	9494002138		1,0816	0,1016	0,0037	0,0845	1,4640
směrodatná odchylka	97437,17021		1,0400	0,3187	0,0609	0,2907	1,2100

Obr. 4.5: Tabulka statistických hodnot na sondě M1 ze vzorku parse.

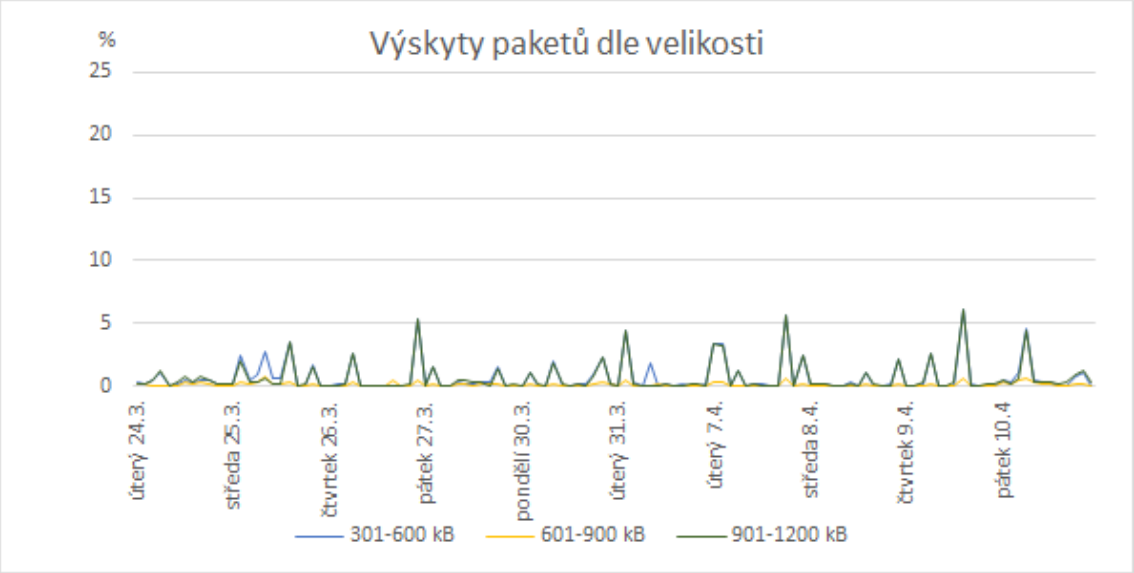
Na M1 provozu se nejčastěji vyskytují pakety o velikosti 1201–1500 kB v hodnotách 60–80 % provozu. Následují pakety o velikosti 0–300 kB, kterých se v provozu nachází 20–40 %. Zbylé intervaly velikostí paketů se pohybují pod 10 %.

Charakteristiku velikostí paketů v provozu M1 by mělo být možné využívat pro detekci anomálií. Ať z pohledu intervalů velikostí, nebo když se poměří výskyty paketů o velikostech 0–300 kB a 1201–1500 kB (obr. 4.6). Zároveň by bylo možné

stanovit maximální hranice výskytu paketů o velikostech 301–600, 601–900 a 901–1200 kB a zvýšený výskyt případně také detekovat jako anomálii (obr. 4.7).



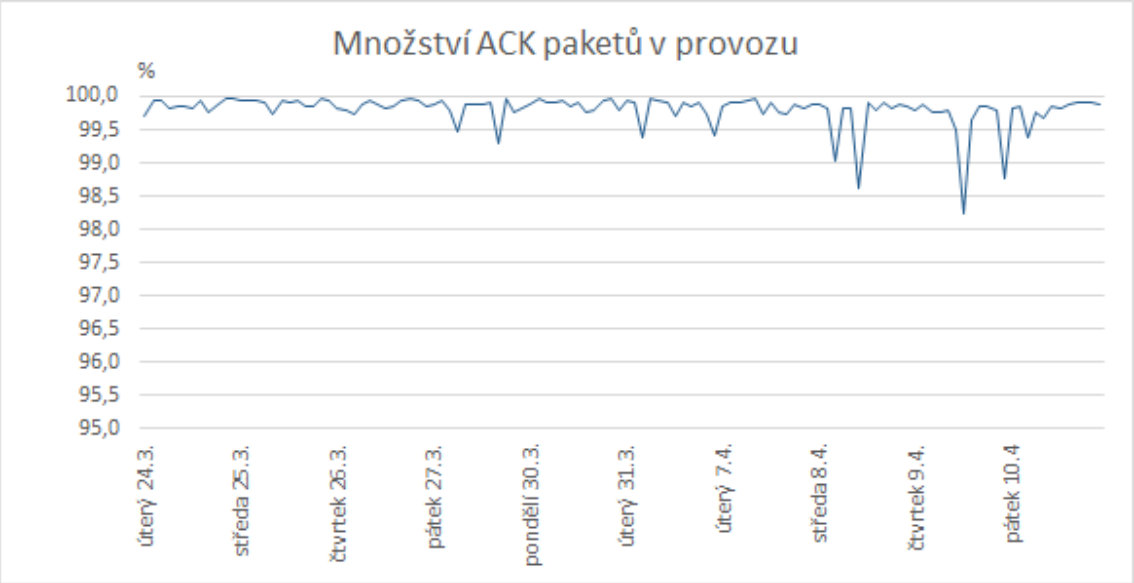
Obr. 4.6: Graf výskytů paketů o velikostech 0–300 a 1201–1500 kB – sonda M1, minutový režim.



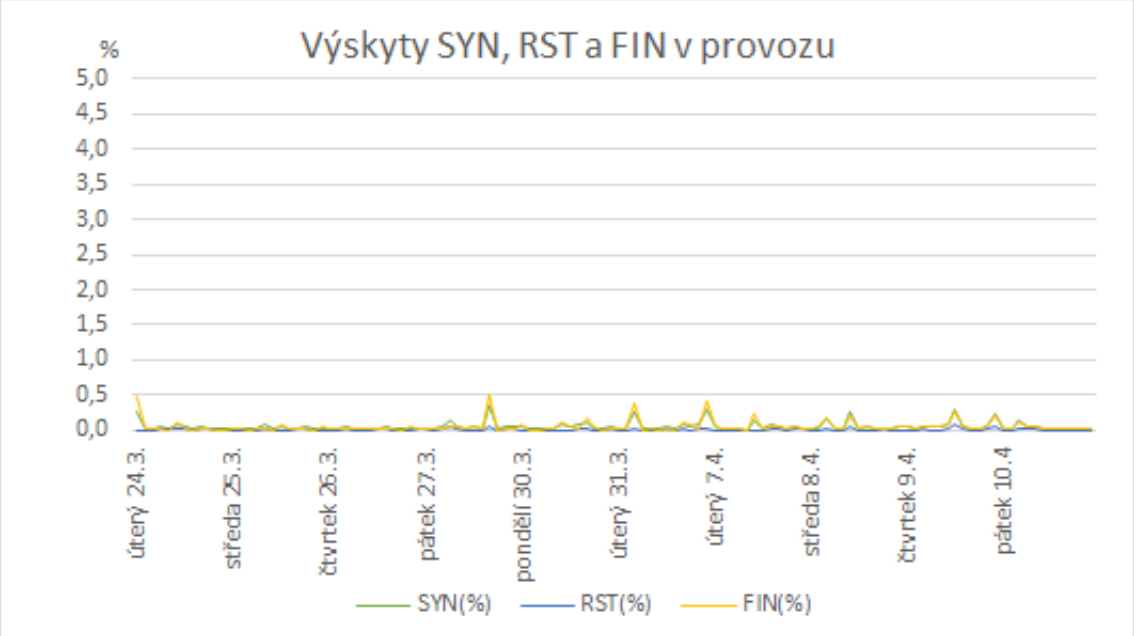
Obr. 4.7: Graf výskytů pakteů o zbylých velikostech – sonda M1, minutový režim.

Pomocí analýzy dat získaných skriptem *flags.ksh* bylo zjištěno, že v zaznamenaném provozu se nachází naprosté minimum s příznaky SYN, RST a FIN (obr. 4.9).

Naopak paketů s TCP příznakem ACK v hlavičce se v provozu nachází více jak 98 % (obr. 4.8).



Obr. 4.8: Graf výskytu ACK příznaku – sonda M1, minutový režim.

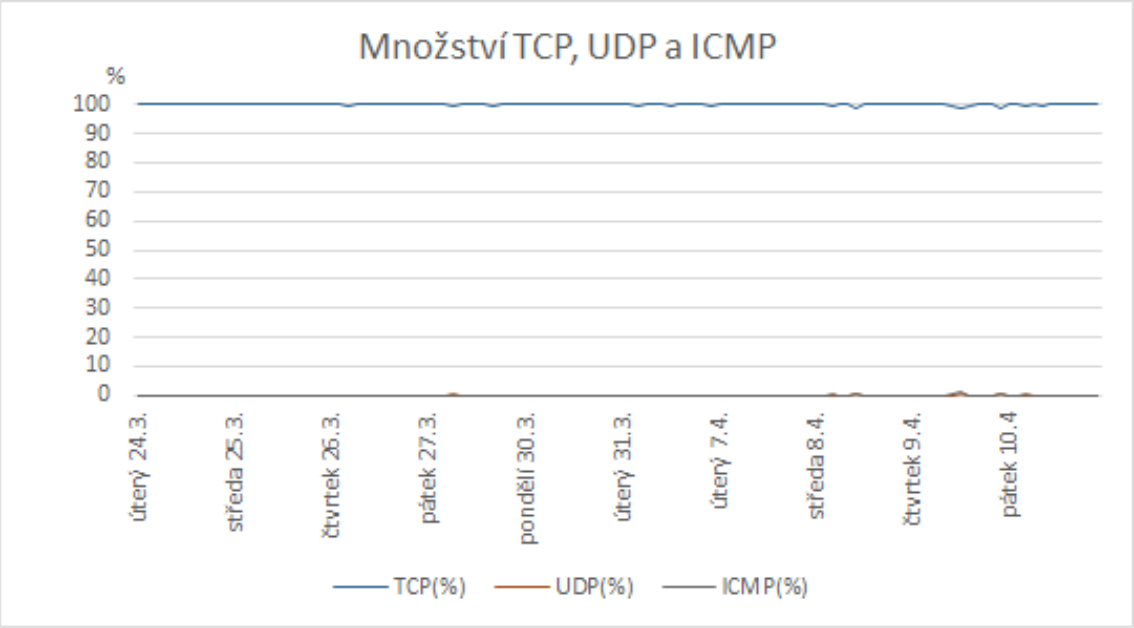


Obr. 4.9: Graf výskytů příznaků SYN, RST a FIN – sonda M1, minutový režim.

Analýza dat získaných pomocí skriptu *ports.ksh* zobrazila, že provoz je dost stereotypní s minimálním výskytem provozu přes sledované porty. Byly detekovány

pouze ojedinělé výskyty komunikace přes HTTP, SSH a DNS s maximální špičkou pod 1 % provozu.

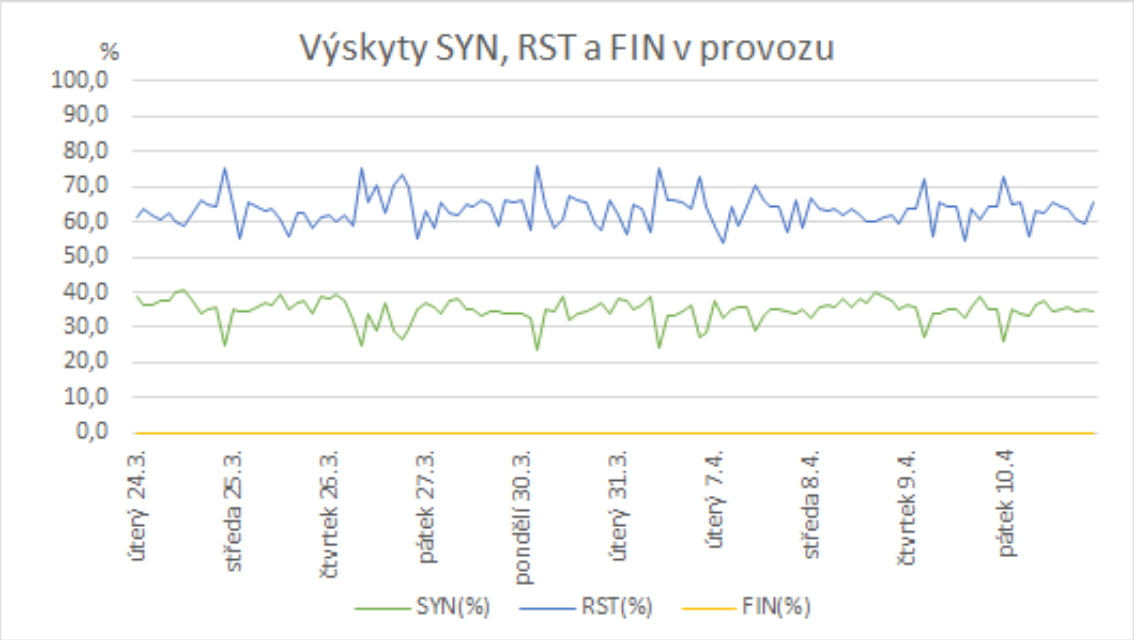
Ze statistického pohledu jsou nejzajímavější hodnoty získané pomocí skriptu *protocol.ksh*. Je zde kontinuální výskyt ICMP a UDP paketů s hodnotou kolem 1 % (obrázek 4.10), což lze sledovat jako stabilní hodnotu bez velkých výkyvů. Naopak je zde trvalý téměř 100 % výskyt paketů TCP. Tyto charakteristiky by mělo být možné sledovat pro případné výskyty anomálií.



Obr. 4.10: Graf provozu sledovaných protokolů na sondě M1 v minutovém režimu.

Na sondě M2, která zaznamenávala provoz mezi firewallem a servery, je již provoz modifikovaný. Téměř celý provoz je pokryt pomocí TCP streamů, aniž by byly použity běžné protokoly a porty stejně jako u sondy M1. Grafy sestavené z dat ze skriptu *parse.ksh* mají podobné průběhy jako u M1. Oproti záznamům ze sondy M1 byl na sondě M2 naměřen relativně stabilní výskyt paketů s TCP příznaky SYN a RST (obr. 4.11) a výrazně nižší výskyt ACK příznaků (obr. 4.12).

Nejzajímavější z pohledu provozu, zaznamenaných dat a časových řad je sonda M3, kde se pravděpodobně určitou anomálii podařilo zaznamenat. Během provozu došlo ke skokovému navýšení přenosu paketů větších jak 1501 KB, které trvalo pouze krátký okamžik a během zaznamenaného provozu se v takové míře již neopakovalo (obr. 4.13).



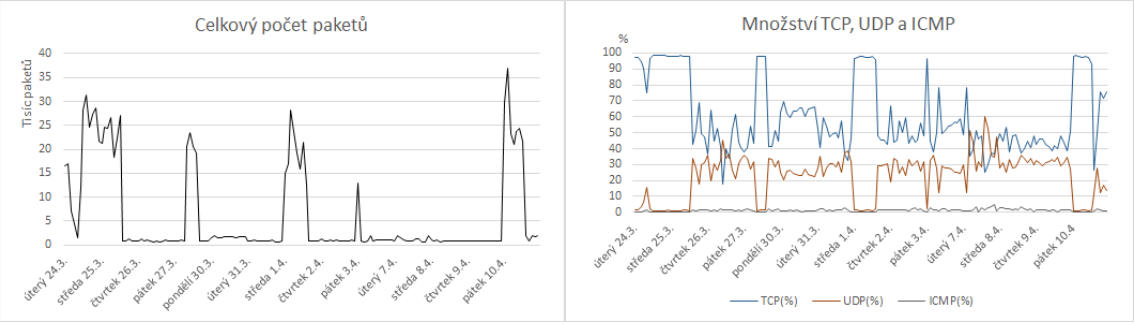
Obr. 4.11: Graf zaznamenaných SYN, RST a FIN flagů – sonda M2, minutový režim.

Obr. 4.12: Graf vyskytu ACK příznaků v minutovém provozu na sondě M2.

Z pohledu časových řad a grafů jsou hodnoty získané skriptem *ports.ksh* na sondě M3 dost rozkolísané. Na známých portech se nachází průměrně 13,2 % TCP provozu a 7,9 % UDP provozu, z čehož 3,6 % pokrývá SSH, 2,1 % DNS, 7,7 % HTTP a 1,5 % HTTPS. V těchto charakteristikách se nepodařilo vysledovat žádnou závislost a pro sledování anomáliií nejsou vhodné.

Obr. 4.13: Graf výskytů paketů dle velikosti na sondě M3 s výrazným výkyvem výskytu paketů o velikosti větší jak 1501 kB.

Ve výstupech ze skriptu *ports.ksh* bylo velmi zajímavé sledovat závislost mezi celkovým provozem a výskytem TCP a UDP paketů, jak je zobrazeno na grafech 4.14).



Obr. 4.14: Grafy zobrazující vzájemný vliv objemu provozu na hodnoty výskytu TCP a UDP paketů.

Je zde vidět, že nárůst přenášených paketů výrazně ovlivňuje celkovou charakteristiku provozu. Navíc k nárůstům dochází pouze s využitím protokolu TCP. Pro sledování tohoto chování by bylo vhodnější dlouhodobé sledování v 24hodinovém režimu. Pro takto krátké období se velmi těžko hledají závislosti a pokud by byly stanoveny určité závěry, tak by se jednalo pouze o spekulace a hrubé odhady.

Dále je vidět, že hodnoty výskytu ICMP paketů v provozu mají téměř konstantní charakter a tento provoz by bylo vhodné nadále monitorovat.

4.5.2 Popisná statistika

Pro všechny naměřené vzorky bylo provedena také analýza za pomocí popisné statistiky. Pro každý datový vzorek byly dopočítány hodnoty koeficientů popsaných v kapitole 3.2. Zároveň byly vytvořeny histogramy četnosti, P-P grafy a krabičkové grafy pomocí rozšíření QI Macros 2015[5] pro MS Excel. Soubory jsou uložené v elektronické příloze ve složce Popisná statistika.

Jako první kritérium byly sledovány hodnoty variačního koeficientu. Vzorky, u nichž je tento koeficient menší než 50 %, byly v tabulkách zvýrazněny a blíže prozkoumány. Jako další hodnoty byly sledovány koeficienty šikmosti, pro zjištění, jak moc se daná hodnota liší nuly a tedy od normálního rozdělení. Pokud se ve sledovaném souboru vstupních dat nacházel vzorek, který splňoval nebo se blížil k těmto podmínkám, tak byly ještě ověřovány podmínky normality dle histogramu četnosti a P-P grafu.

Při kladném vyhodnocení je možné statistické hodnoty tohoto vzorku využít jako výchozí hodnoty pro případnou analýzu reálného datového provozu. Vždy je třeba počítat s tím, že různé provozy mají různé charakteristiky a získané hodnoty lze použít pouze na konkrétní sondu, ze které byla zdrojová data získána.

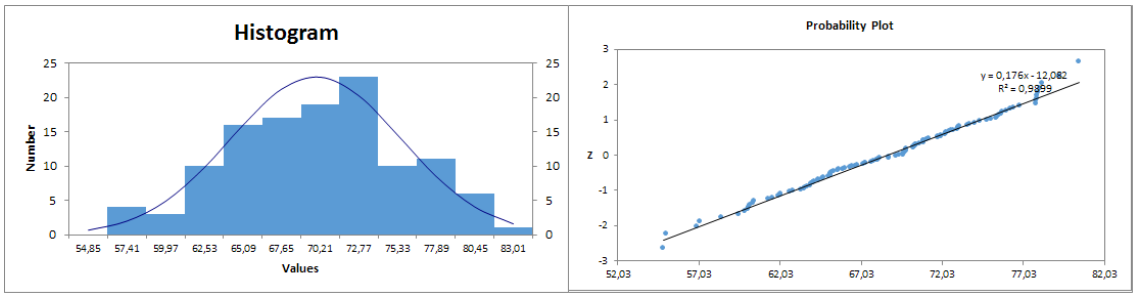
Popisná statistika						
	0-300 kB	301-600 kB	601-900 kB	901-1200 kB	1201-1500 kB	total
Minimum	23,51137	0,00864	0,00303	0,00558	53,85761	105711,00000
Q1	33,85100	0,03541	0,01827	0,06472	60,70194	195769,75000
Medián - Q2	35,20911	0,06114	0,04249	0,12930	63,75529	406381,00000
Q3	36,85378	0,21686	0,12074	0,44745	65,37181	787313,75000
Maximum	40,82756	6,42737	0,72966	6,44935	76,21825	2415284,00000
Průměr	34,75415	0,83702	0,11633	0,87406	63,41844	561496,69167
Rozptyl	11,28064	2,44925	0,02489	2,30382	19,66454	212245670099,48000
Sm.odchylka	3,35867	1,56501	0,15777	1,51783	4,43447	460701,28077
Variační koef.	9,66407	186,97275	135,62940	173,65402	6,99240	82,04880
Koef. šikmosti	-1,40973	1,88484	1,88214	1,97637	0,54655	1,63257
Koef. špičatosti	2,36896	2,35082	2,89963	2,83528	0,88456	3,12740

Obr. 4.15: Tabulka statistických hodnot získaných skriptem *parse.ksh* na sondě M1.

[5]Využito 30denní zkušební verze, staženo z http://www.qimacros.com/

Na sondě M1 bylo zjištěno téměř normální rozdělení u 2 vzorků získaných skriptem *parse.ksh*. Sledovaným podmínkám vyhovovaly výskyty paketů o velikosti 0–300 kB a 1201–1500 kB, kde variační koeficient dosahuje hodnot menších než 10 % a zároveň hodnoty koeficientu šikmosti jsou velmi blízké nule. V tabulce 4.15 jsou hodnoty vyhovujících variačních koeficientů zvýrazněny.

Následně byla tato sledování prověřena kontrolou histogramu četnosti a P-P grafu.



Obr. 4.16: Histogram a P-P graf pro výskyty paketů o velikosti 1201–1500 kB na sondě M1.

Grafy 4.16 potvrdily, že u těchto vzorků lze počítat s přibližně normálním rozdělením a lze tedy dále pracovat s naměřenými a dopočítanými hodnotami. Pro další výpočty je vhodné použít pravidla tří sigma, které říká, že všechny hodnoty statistického souboru v normálním rozdělení by se měly nacházet maximálně do tří odchylek od průměru.[24]

Pokud by byla tato metoda použita v reálném provozu, tak by se nově měřené hodnoty měly nacházet v intervalech:

- $(\overline{y} - s_y; \overline{y} + s_y)$ s pravděpodobností 68,27 %,
- $(\overline{y} - 2s_y; \overline{y} + 2s_y)$ s pravděpodobností 95,45 %,
- $(\overline{y} - 3s_y; \overline{y} + 3s_y)$ s pravděpodobností 99,73 %.

Pro potřeby reálného provozu by měly být dostačující první dva intervaly s tím, že překročení prvního intervalu může sloužit jako hranice pro zobrazení žlutého alarmu pro varování mírné odchylky od standardního provozu a překročení druhého by značilo výrazný výkyv od standardního provozu, tedy červený alarm.

Tímto způsobem byly prověřeny a zhodnoceny všechny získané vzorky na každé ze sond. Z ostatních měření na sondě M1 by dle variačního koeficientu mělo být možné dále pracovat i hodnotami výskytů paketů s TCP příznaky ACK a SYN/ACK nebo

výskyty TCP paketů. U všech těchto vzorků je však hodnota koeficientu šikmosti nebo špičatosti výrazně odchýlená od potřebných hodnot. Ostatní vzorky mají asymetrické rozdělení a nelze je tedy pro další výpočty v takovémto stavu využít. Dle analýzy histogramů četnosti by rozdělení většiny zbylých vzorků mohlo odpovídat logaritmickému nebo exponenciálnímu rozdělení. Pokud by s těmito vzorky bylo potřeba dále pracovat, je možné provést jejich transformaci, vykonat potřebné výpočty a poté zpětnou transformací získat hledané hodnoty průměru a odchylky. V reálném provozu by tento postup nebylo vhodné používat z důvodu rostoucí výpočetní náročnosti a při analýzách by mohlo docházet ke zpožďování.

Na sondě M2 je statistické vyhodnocení výstupu ze skriptu *parse.ksh* téměř identické s hodnotami na sondě M1. Variační koeficienty pro výskyty paketů o velikostech 0–300 a 1201–1500 kB jsou velmi nízké. Avšak hodnoty výskytu paketů 0–300 kB již jsou výrazněji zešikmené směrem doprava, takže by nebylo vhodné bez další úpravy tuto charakteristiku sledovat.

Další výpočty však přinesly zajímavé zjištění. V kapitole 4.5.1 byla zmíněna domněnka, že provoz, který sonda M2 zaznamenala, je nejspíš tvořen TCP streamy. Ze statistických výpočtů vyšlo téměř normální rozdělení výskytu TCP paketů s průměrem 99,6425 % a směrodatnou odchylkou 0,2620, což značí, že téměř celý provoz je zajištěn TCP pakety. V tomto případě by zvýšený výskyt jiných paketů než TCP mohl značit anomálii, např. záplavu UDP nebo ICMP, ale nic takového se detekovat nepodařilo. Další charakteristika, která má téměř normální rozdělení, je výskyt paketů s příznaky PSH/ACK, tedy signalizace probíhajících TCP streamů.

Z pohledu popisné statistiky a použití základních statistických souborů získaných ze záznamů není na sondě M3 žádný soubor, který by bylo možné využít pro další výpočty. Ačkoliv má několik charakteristik dobré hodnoty variačních koeficientů, případně vyhovující hodnoty koeficientu šikmosti, tak při kontrole histogramů četnosti a P-P grafů bylo zjištěno, že rozdělení není normální.

4.6 Testovací analýza

Ještě před samotnou analýzou datového provozu byly zvažovány možnosti ověření funkčnosti statistických metod. Bylo uvažováno, že v běžném provozu by měl být zhruba stejný výskyt nejčastěji používaných portů ze skupiny známých portů. Proto byl vytvořen skript *ports_an.ksh* (příloha A.6) pro analýzu provozu, který by byl modifikován pro ověření, zde se podaří tuto úpravu detekovat jako anomálii.

Předpokládalo se, že objem provozu na sledovaných portech by měl být kolem 95 % z provozu na známých portech.

Logika skriptu je založena na tom, že jsou vypočítané, a tedy i dané, hodnoty průměru a směrodatné odchylky pro výskyty sledovaných paketů v provozu omezeném na známé porty. Výše bylo zmíněno, že pro potřeby detekce anomálií postačuje pravidlo 2s. Pomocí těchto hodnot lze nastavit hladinu standardního provozu, který je roven aritmetickému průměru výskytu sledovaných paketů v provozu známých portů, hodnota pro zobrazení žlutého alarmu je aritmetický průměr – směrodatná odchylka a hranice pro červený alarm je rovna průměru – 2 směrodatné odchylky. Při tvorbě skriptu byly zvoleny hodnoty 95 % pro standardní provoz, 85 % pro hranici žlutého alarmu a 75 % pro červený. V současnosti lze skript použít pouze pro analýzu již zaznamenaného provozu.

Skript pro svůj běh využívá externí soubor *porty.txt*, ve kterém jsou vypsána čísla sledovaných portů:

- 22 – SSH,
- 25 – SMTP,
- 53 – DNS,
- 80 – HTTP,
- 143 – IMAP,
- 443 – HTTPS,
- 465 – SMTPS,
- 993 – IMAPS.

Po spuštění provede skript součet paketů na známých portech ve vstupním souboru *a.pcap* a zároveň si pro úsporu času tento provoz vyfiltruje a uloží do nového souboru *b.pcap*. Dokud nejsou nejsou zpracovány všechny řádky a tedy i porty ze souboru *porty.txt*, probíhá v cyklu analýza souboru *b.pcap* a počítají se výskyty paketů na daných portech:

```
while read radek; do
P=‘tshark -r b.pcap -Y "tcp.port==$radek or udp.port==$radek" | wc -l‘
x=$[$predchozi+$P]
predchozi=$x
done < porty.txt
```

Po ukončení cyklu je proveden výpočet, jak velkou část provozu na známých portech pokrývá provoz sledovaných portů a získaná hodnota je porovnána s hodnotami danými v úvodu.

```
pomer=$[$x * $base / $T]
echo "Pomer namerenych a celkovych paketu je $pomer %"
```

Na základě porovnání je zobrazeno hlášení, zda je sledovaný provoz v pořádku, nebo zda byla detekována anomálie vůči zadání.

Provoz, který byl v rámci této práce zpracováván, však pro tuto konkrétní analýzu není vhodné použít. Datový soubor sledovaného provozu na známých portech má na sondách M1 a M2 vysoké hodnoty variačního koeficientu a zároveň jsou zde velké rozdíly mezi minimální a maximální hodnotou, což má za následek velké hodnoty rozptylů a tím i směrodatných odchylek. Pokud by se na tyto hodnoty použilo pravidlo dvou sigma, tak by hladiny alarmů byly mimo rozmezí 0–100 %.

4.6.1 Testování s uměle vytvořeným provozem

Z již provedených měření a výpočtů by pro podobnou detekci anomálií měly být vhodné například výskyty paketů o velikostech 0–300 nebo 1201–1500 kB. Proto byl vytvořen nový skript, *parse_an.ksh* (v příloze A.7), aby mohl sledovat a vyhodnocovat hladinu výskytů paketů o velikosti 0–300 kB v provozu na sondě M1. Z předchozích statistických výpočtů byly získány pro definici vstupních proměnných tyto hodnoty (zaokrouhleno na celá čísla):

- aritmetický průměr – 30,
- směrodatná odchylka – 5.

Jak již bylo popsáno v předchozím textu, tak s pomocí aritmetického průměru a směrodatné odchylky je možné určit hraniční hodnoty pro vyvolání alarmu. Pokles nebo nárůst sledovaných dat o jednu odchylku definuje hranice pro žlutý alarm a o dvě odchylky jako červený alarm. Ve skriptu je to zajištěno pomocí proměnných pro dolní a horní hranici žlutého a červeného alarmu:

```
#deklarace proměnných získaných z analýzy provozu
    avg=30 #průměr
    dev=5 #odchylka
#nastavení hranic alarmů
    yeld=$[$avg - $dev]
    yelu=$[$avg + $dev]
```

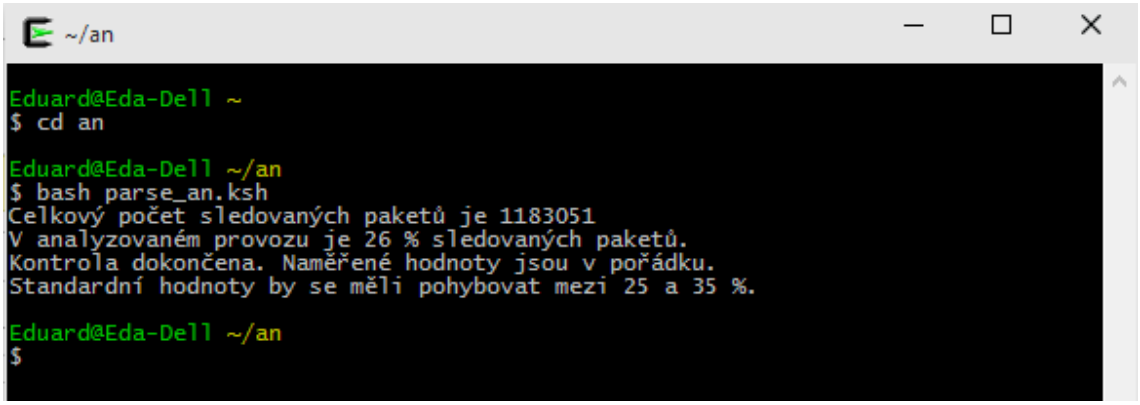
```
    redd=$[$yeld - $dev]
    redu=$[$yelu + $dev]
```

Dále je ve skriptu provedeno porovnání a vyhodnocení naměřených hodnot vůči daným hranicím pro vyvolání alarmu:

```
#porovnáme se vstupními hodnotami
    if [ "$pomer" -lt "$redd" ] || [ "$pomer" -gt "$redu" ];
        then echo -e "\e[31mRed Alarm!!! \e[39mV analyzovaném provozu je
        detekována výrazná odlišnost výskytu paketů o velikosti 0--300 kB
        oproti standardnímu provozu."
    elif [ "$pomer" -lt "$yeld" ] || [ "$pomer" -gt "$yelu" ];
    then echo -e "\e[33mYellow Alarm!!! \e[39mV analyzovaném provozu je
    detekována odlišnost výskytu paketů o velikosti
    0--300 kB oproti standardnímu provozu."
else
        echo "Kontrola dokončena. Naměřené hodnoty jsou v pořádku."
fi
```

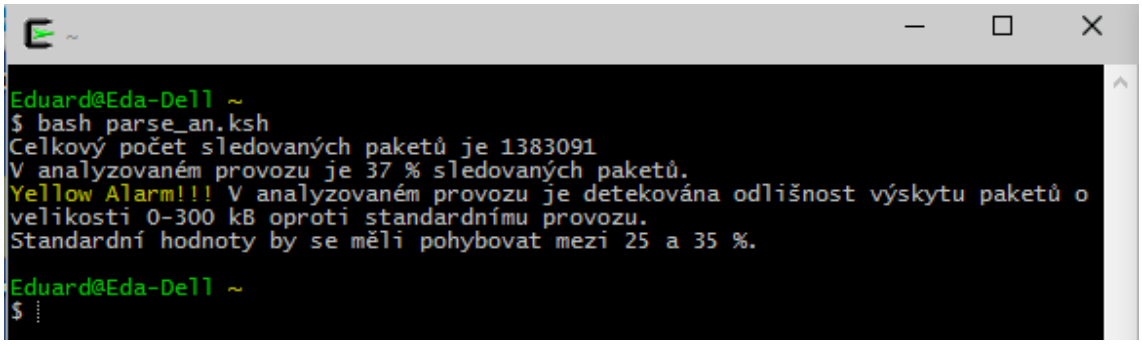
Protože tento skript sleduje a poměřuje pouze hodnoty celkového počtu paketů a paketů o dané velikosti, je výrazně jednodušší a mělo by být možné jej provozovat i v reálném čase.

Pro otestování skriptu byl použit soubor *03-30_00005_20150330090502.pcap*, který byl již analyzován v rámci úvodní analýzy minutových vzorků. Výsledek analýzy tohoto souboru je zobrazen na obrázku 4.17 a hodnoty souhlasí s původními hodnotami, které byly získány pomocí skriptu *parse.ksh*.



Obr. 4.17: Snímek po vykonání testovacího skriptu na zdrojovém souboru.

Aby bylo možné funkčnost skriptu otestovat a zároveň ověřit vypočtené statistické hodnoty, bylo třeba originální provoz modifikovat. Proto byl pomocí aplikace Ostinato[6], což je generátoru datového provozu, vytvořen *pcap* soubor, který obsahuje záznam přenosu 100000 paketů o náhodné velikosti 64–300 kB a 20 paketů pro režii. Vygenerovaný provoz byl následně pomocí funkce Merge ve Wiresharku spojen se zdrojovým provozem a výstupní soubor byl zanalyzován, zda je možné tento nárůst přenosu paketů detekovat.



Obr. 4.18: Žlutý alarm po analýze původního *pcap*u obohaceného o 200000 paketů.

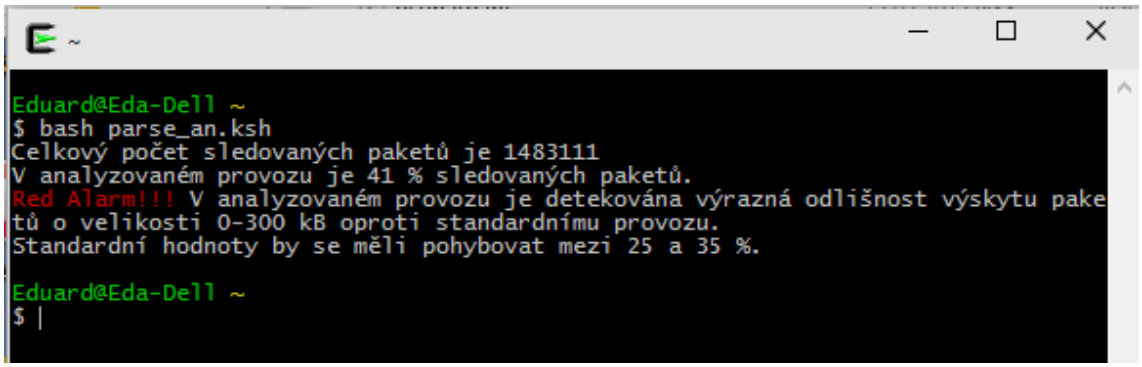
Při analýze provozu obohaceného o vygenerovaný záznam 100000 paketů nebyly zjištěny žádné anomálie. Přestože došlo k nárůstu objemu sledovaných paketů, naměřená hodnota byla stále v hodnotách, které odpovídají limitům standardního provozu.

Aby bylo možné případné anomálie detekovat, byl původní objem paketů opakovaně navyšován, vždy o 100000 paketů z uměle vygenerovaného provozu. Po každém navýšení byla provedena analýza pomocí výše uvedeného skriptu pro ověření vlivu na sledované charakteristiky provozu.

Po navýšení o 200000 paketů nedošlo k očekávanému vyvolání alarmu pro výraznou anomálii ve sledovaném provozu. Byl detekován pouze žlutý alarm. Původní minutový záznam bylo nutné celkem třikrát obohatit o vygenerovaný provoz tak, aby hodnota výskytu paketů o velikosti 0–300 kB překročila hranici pro červený alarm.

Z tabulky 4.20 je patrné, že pro detekci anomálie na sledovaném výskytu paketů o velikostech 0–300 kB, bylo nutné navýšení množství paketů této velikosti alespoň o 60 % pro žlutý alarm a o 100 % pro červený alarm. Ačkoliv se tato hodnota může

[6]Volně dostupné na adrese https://code.google.com/p/ostinato/wiki/UserGuide



Obr. 4.19: Červený alarm po analýze původního *pcap*u obohaceného o 300000 paketů.

zdát vysoká, tak v celkovém množství přenesených paketů došlo k nárůstu pouze o 25 %. Navíc při analýze časových řad (kapitola 4.5.1) bylo zjištěno, že na sondě M1 může provozní špička dosáhnout až 1800000 paketů/min a touto trojitou modifikací původního provozu došlo k navýšení pouze na necelých 1500000 paketů/min. Tedy z pohledu časové řady by takovýto provoz jako anomálie vůbec nevypadal, ale při detailní analýze s pomocí popisné statistiky již lze potvrdit, že se v uměle modifikovaném provozu nachází výrazná anomálie.

Testovací analýza pro výskyt paketů 0-300 kB na sondě M1				
	Celkový počet paketů	Počet paketů 0-300 kB	Výskyt sledovaných paketů	Detekována anomálie?
Zdrojový soubor	1183051	317577	26,84%	ne
+100020 paketů	1283071	417597	32,55%	ne
+200040 paketů	1383091	517617	37,42%	ano - žlutý alarm
+300060 paketů	1483111	617637	41,64%	ano - červený alarm

Obr. 4.20: Souhrnný přehled testovací analýzy s uměle vyvolanou anomálií.

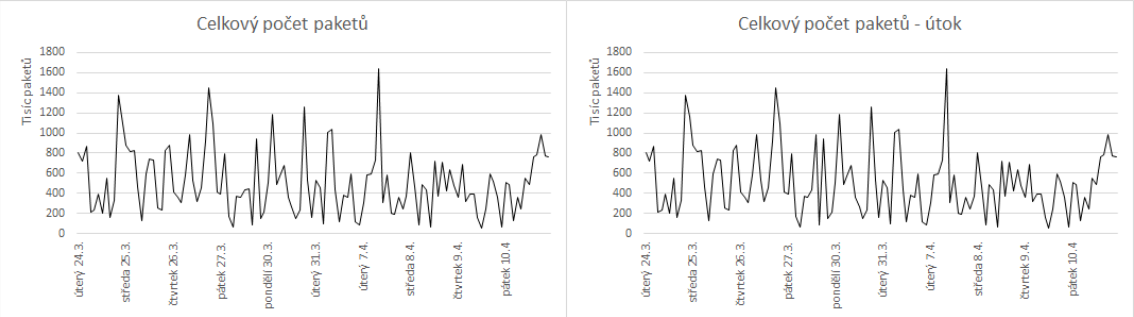
4.6.2 Testování útoku typu záplava SYN

Již bylo potvrzeno, že s využitím provedených měření a statistických výpočtů lze v datovém provozu detekovat anomálie. Doposud se jednalo o teoretické úvahy a cílenou modifikaci provozu. Jakým způsobem se změní charakteristika provozu, pokud dojde k reálnému útoku? A lze z případné detekované anomálie určit typ útoku?

Z internetu[7] byl stažen *pcap* soubor reálného útoku typu záplava SYN (1.3.2). Při náhledu na tento soubor ve Wiresharku bylo zjištěno, že se jedná o záznam o délce

[7]Staženo z https://github.com/somethingnew2-0/CS642-HW2/blob/master/traces/synflood.pcap

pouze 9 vteřin, který obsahuje 539420 paketů. Tímto souborem byl modifikován jeden náhodně zvolený minutový záznam na sondě M1 a poté byly provedeny analýzy z pohledu časových řad. Výsledky byly porovnány s původním čistým provozem.



Obr. 4.21: Porovnání grafů celkových přenesených paketů.

Nejprve bylo provedeno vzájemné porovnání dat získaných skripty *parse.ksh*. Mezi grafy sestavenými z celkového počtu paketů v jednotlivých souborech není znatelný téměř žádný rozdíl. Navýšení přenášených dat o cca 500000 paketů v rámci jednoho souboru není, tak velká změna, aby ji bylo možné odhalit při porovnání grafů původního a modifikovaného provozu (obr. 4.21). V další charakteristice z tohoto skriptu je vidět krátkodobý výrazný nárůst množství paketů o velikosti 0–300 kB, ve špičce dojde k nárůstu až na 70 % z celkového objemu provozu. Oproti běžnému provozu, kdy se tato hodnota pohybuje mezi 20 a 40 %, je nárůst téměř dvojnásobný (obr. 4.22).

Vzhledem k tomu, že je nárůst pouze krátkodobý, bylo jej možné zaznamenat pouze v detailním grafu s minutovými záznamy. Grafy, kde jsou denní provozy zprůměrovány, nevykazují žádný rozdíl.

Jako další bylo provedeno porovnání grafů z dat získaných skriptem *flags.ksh*, který zjišťuje výskyty TCP příznaků v analyzovaných souborech. V původním provozu je množství paketů s příznaky ACK dlouhodobě u hranice 100 % (obr. 4.8), nejnižší hodnoty dosahují 98 %. V průměrném zobrazení má téměř kontinuální průběh. Oproti tomu graf modifikovaného průběhu vykazuje v minutovém režimu pokles výskytu ACK flagů až k hranici 40 % (obr. 4.23). Navíc je tento pokles znatelný i v zprůměrovaném denním režimu (obr. 4.25).

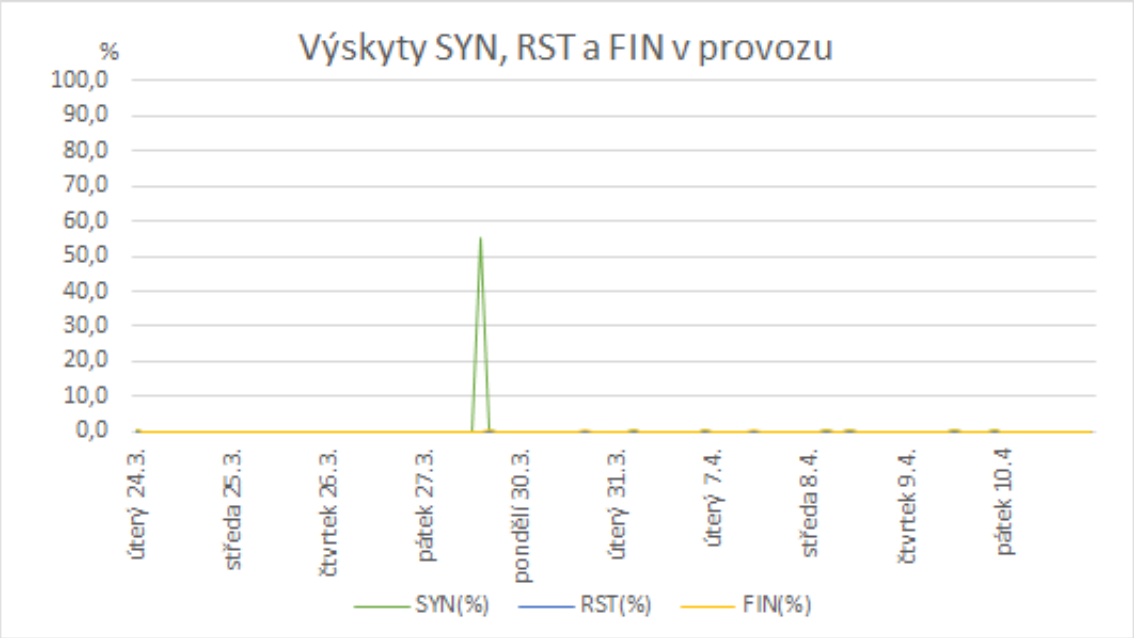
Další graf zobrazuje výskyty příznaků SYN, RST a FIN v provozu. V běžném provozu se hodnoty výskytů paketů s těmito příznaky pohybují pod 1 % (obr. 4.9)

Obr. 4.22: Grafy zobrazující přenosy paketů o velikostech 0–300 a 1201–1500 kB.

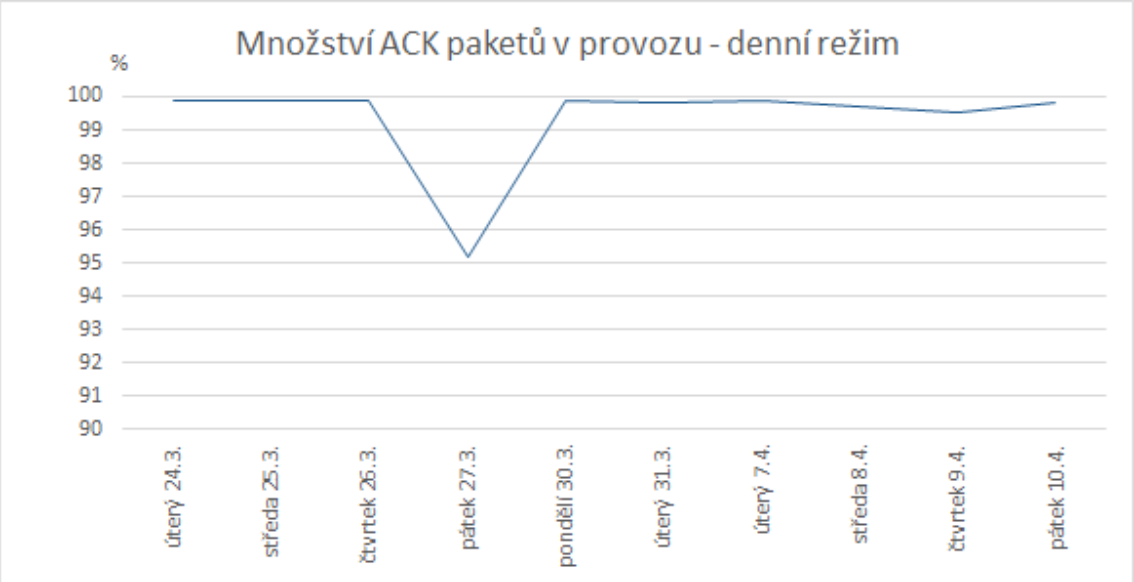
Obr. 4.23: Graf zaznamenaného poklesu paketů s příznaky ACK – minutový režim.

a v provoze, který byl modifikován, byl zaznamenán nárůst příznaků SYN a k hranici 60 % (obr. 4.24), což je opravdu extrémní nárůst a byl detekován i denním režimu (obr. 4.26.

Grafy sestrojené z dat získaných zbylými skripty *ports.ksh* a *protocol.ksh* mají velice podobné průběhy jako grafy průvodního provozu. Porovnání původního a nového průběhu při těchto kontrolách neprokázalo žádnou významnou odchylku.



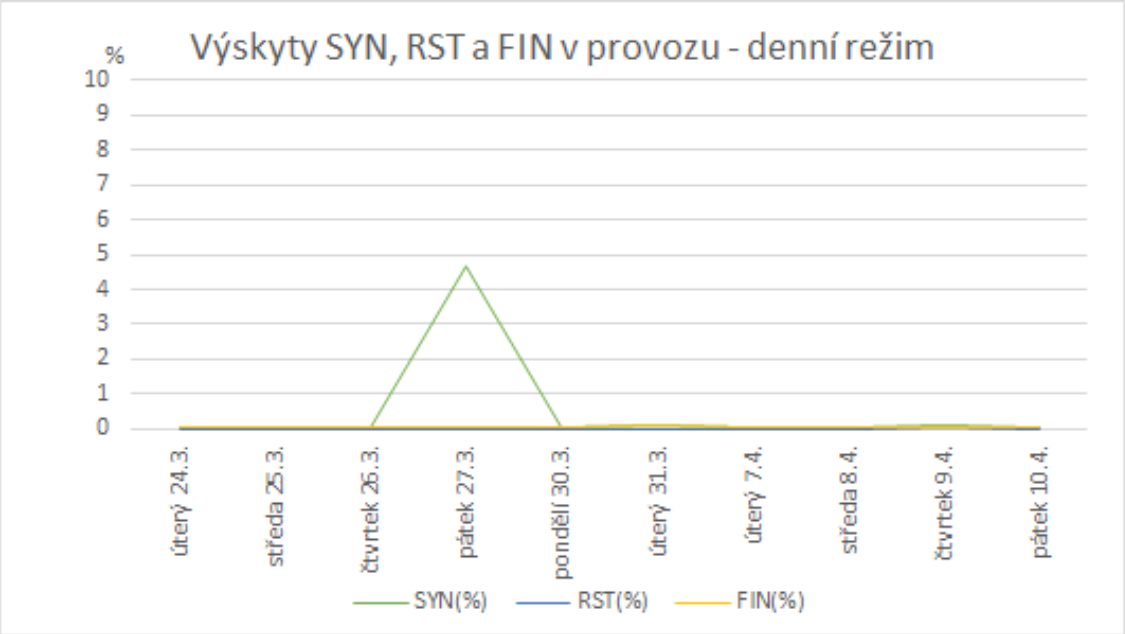
Obr. 4.24: Graf s viditelným nárůstem výskytu paketů s příznaky SYN – minutový režim.



Obr. 4.25: Graf zaznamenaného poklesu paketů s příznaky ACK – denní režim.

Z porovnání obou provozů tedy vyplynulo několik významných rozdílů, které byly krátkodobě detekovány:

- navýšení přenosu malých paketů o velikosti 0–300 kB
- pokles paketů s TCP příznakem ACK
- nárůst TCP příznaků SYN, aniž by narostlo množství SYN/ACK



Obr. 4.26: Graf zaznamenaného nárůstu paketů s příznaky SYN – denní režim.

Zároveň byl proveden test analytického skriptu, který zaznamenal extrémní hodnotu paketů o velikosti 0–300 kB v minutovém vzorku (obr. 4.27).

```
Eduard@Eda-Dell ~/a
$ bash parse_an.ksh
Celkový počet sledovaných paketů je 980065
V analyzovaném provozu je 68 % sledovaných paketů.
Red Alarm!!! V analyzovaném provozu je detekována výrazná odlišnost výskytu pake
tů o velikosti 0-300 kB oproti standardnímu provozu.
Standardní hodnoty by se měly pohybovat mezi 25 a 35 %.
```

Obr. 4.27: Snímek po provedení analytického skriptu napadeného provozu.

Ze zjištěných sledování bylo vyhodnoceno, že se jednalo o krátkodobý, ale velice silný nárůst množství paketů o malé velikosti s TCP příznakem SYN. Z toho lze usuzovat, že by se mohlo jednat o útok SYN flood, což potvrzuje typ útoku, kterým byl provoz modifikován.

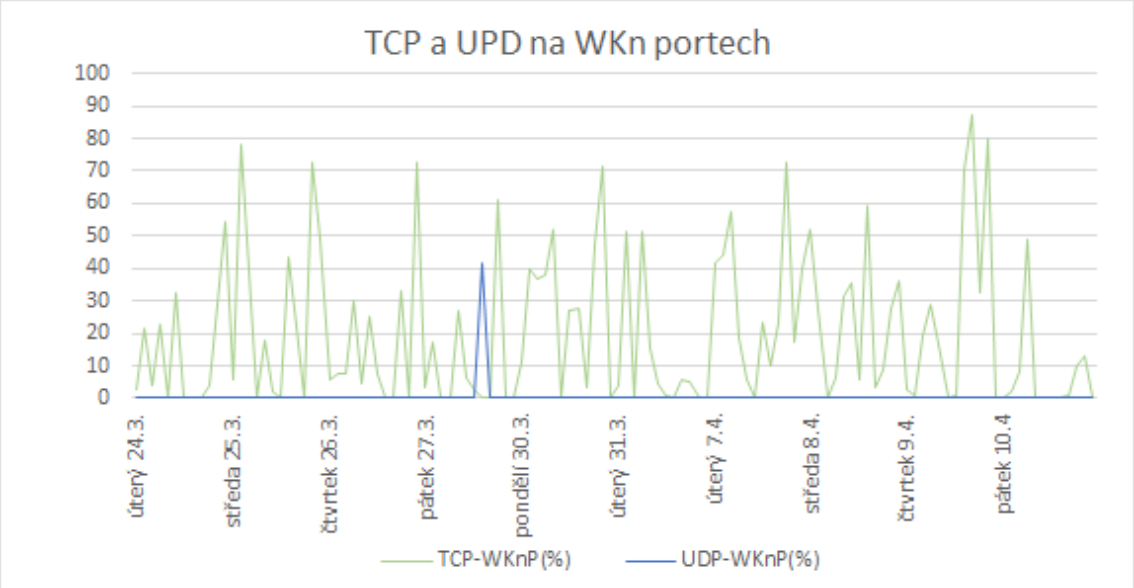
4.6.3 Testování útoku typu na DNS

Pro další test byl stažen soubor s útokem na službu DNS[8]. Záznam měl celkovou délku větší než 6 minut, proto byl rozdělen na minutové úseky. Stejně jako u SYN

[8]Staženo z http://www.simpleweb.org/wiki/Traces

flood útoku byl minutový záznam útoku připojen k náhodnému minutovému záznamu původního provozu. Protože minutový záznam DNS útoku obsahuje více jak 300000 paketů, je možné jím modifikovat pouze provoz na sondě M1 nebo M2. Na sondě M3 by takovýto nárůst bylo možné detekovat okamžitě vzhledem k několikanásobnému nárůstu přenášených paketů.

Byly provedeny totožné testy jako v předchozím případě. Opět byl detekován nárůst přenosu paketů o velikosti 0–300 kB a pokles paketů s ACK příznaky. V dalších testech byly zjištěny výrazné výkyvy na množství TCP a UDP paketů. Byl detekován krátký nárůst hladiny UDP, která se v běžném provozu pohybuje pod hranicí 1 %. V minutové analýze byl zjištěn v provozu výskyt UDP na známých portech, který přesáhl hranici 40 % (obr. 4.28). Zároveň při analýze portů byl detekován abnormální nárůst množství paketů na portu 53, tedy DNS (obr. 4.29).

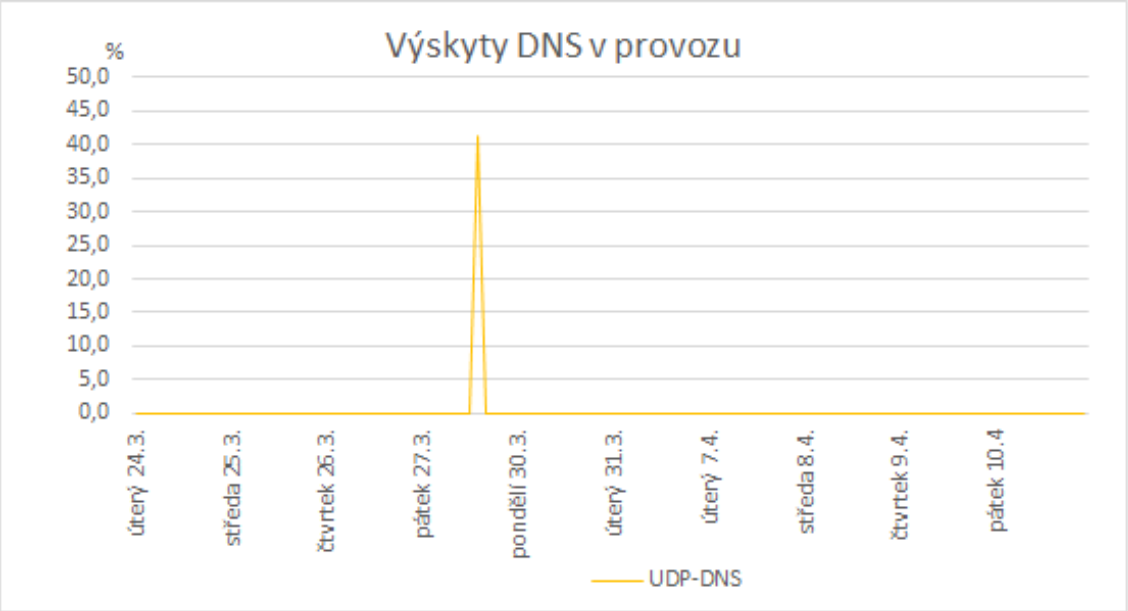


Obr. 4.28: Graf TCP a UDP provozu na známých portech – minutový režim.

V grafech sestavených z minutových průměrů v denním režimu tyto nárůsty nebyly tak znatelné. V tomto režimu byly zjištěny maximální hodnoty UDP a DNS kolem 4 % provozu. Pro detekci v dlouhodobém režimu by bylo třeba, aby byly citlivě nastaveny hranice pro detekci anomálií.

Z testů provedených na tomto vzorku bylo zjištěno:

- Nárůst množství paketů malé velikosti.
- Nárůst UDP provozu.
- Výrazný nárůst množství paketů využívajících port 53 (DNS) v provozu.



Obr. 4.29: Graf zobrazující nárůst DNS v provozu – minutový režim.

V rámci modifikovaného provozu byla detekována anomálie, jejich příznaky by nasvědčovaly možnému útoku na službu DNS pomocí velkého množství zaslaných DNS dotazů.

5 VYHODNOCENÍ

Během provedených analýz byly u všech zaznamenaných provozů sledovány charakteristiky, jejichž výraznější změny by mohly pomoci k případné detekci útoku. U všech vzorků bylo sledováno:

- množství paketů v provozu,
- velikosti přenášených paketů,
- množství různých TCP příznaků v hlavičkách paketů,
- výskyty TCP, UDP a ICMP v provozu,
- komunikace na nejčastěji využívané porty.

Ze zjištěných charakteristik byly sestaveny časové řady a grafy pro ověření, zda by bylo u konkrétního provozu pro případnou detekci anomálií dané chování sledovat.

Na sondě M1 by pro případnou detekci bylo možné využít sledování různých vlastností. V provozu byly naměřeny relativně stabilní hodnoty výskytů paketů o velikostech 0–300 a 1201–1500 kB. Dále bylo zjištěno, že provoz obsahuje v téměř všech paketech příznak ACK. Množství příznaků SYN, RST a FIN se trvale pohybovalo pod hranicí 1 %. Z pohledu portů a protokolů byl na sondě M1 zjištěn téměř nepřetržitý přenos TCP paketů s minimálním výskytem UDP a ICMP provozu a také bez výraznější komunikace na sledovaných portech, které bývají nejčastěji využívány.

U sondy M2 bylo sledováno podobné chování z pohledu velikostí paketů, portů i protokolů. Odlišné byly charakteristiky TCP příznaků. Byly sledovány relativně stabilní hodnoty výskytů příznaků SYN a RST a oproti sondě M1 byly hodnoty příznaku ACK v provozu stabilně na hodnotách nepřesahujících 5 %.

Na sondě M3 nevykazovala žádná z měřených charakteristik takové chování, aby jej bylo možné sledovat pro možný výskyt anomálií. Provoz byl velice různorodý a výskyty jednotlivých sledovaných parametrů natolik rozkolísané, že by jejich využití pro další výpočty nebylo možné. Jediná vlastnost, kterou by bylo možné sledovat, byly hodnoty TCP, UDP a ICMP v provozu. Při delším sledování by mělo být možné stanovit hladiny výskytu TCP a UDP, které by mohly pomoci při případné detekci anomálií. Množství ICMP bylo dlouhodobě nízké stejně jako u předchozích záznamů.

Na základě provedených měření a zjištěných závislostí byly simulovány útoky pomocí záplavy SYN paketů a zahlcení DNS dotazy. Oba útoky se při analýzách modifikovaného provozu podařilo detekovat jako anomálie. Z porovnání sledovaných

charakteristik a známých vlastností útoků bylo odhadnuto, o jaký typ útoku by se mohlo jednat a zjištěné výsledky souhlasily s typem útoku, kterým byl původní provoz infikován.

Pokud by se jednalo o útok typu záplava UDP, tak by je mělo být možné detekovat na sondách M1 a M2, protože při měřeních bylo zjištěno, že výskyty UDP v čistém provozu jsou naprosto minimální. Na sondě M3 byl UDP provoz detekován, a aby případný útok vyvolal anomálii, musel by být veden s velkou razancí.

Poslední typ útoku, který byl zmíněn v teoretické části, je záplava ICMP. Ve všech sledovaných provozech bylo zjištěno, že výskyty ICMP paketů v běžném provozu jsou relativně stabilní na hodnotách blížících se 0 %. Pokud by tedy byl proveden útok toho typu, mělo by být možné jej detekovat na všech sondách jako odchylku od standardního provozu.

6 ZÁVĚR

Cílem této práce bylo prozkoumat a vyhodnotit, zda je možné využít statistické metody pro detekci anomálií v datovém provozu.

V teoretické části byly detailně popsány nejčastější druhy datových útoků se zaměřením hlavně na útoky typu DDoS, které by mohly charakteristiku datového provozu ovlivnit takovým způsobem, že by se při analýze mohly projevit jako anomálie. Zároveň byla popsána metodika vedení těchto útoků a dále byly nastíněny možnosti prevence.

V další kapitole byly zvolené útoky zanalyzovány z teoretické stránky, zda jejich přítomnost v datovém provozu může být detekována podle určitých rysů, které byly dále sledovány v reálném provozu. Součástí teoretické části je i popis základních statistických metod a parametrů, které byly v rámci práce použity pro vlastní analýzu reálných vzorků.

Praktické řešení je zahájeno detailním popisem, kde a v jakém období byla data sbírána. Dále byla popsána metodika, jak byla zdrojová data upravována pro další analýzu. Součástí řešení byla i analýza náhodného denního provozu, která posloužila k výběru určité hodiny, která by z pohledu provozu měla být tou nejvytíženější. Za pomocí vlastních vytvořených skriptů byly ze zdrojových souborů získána data, která byla následně zanalyzována z pohledu časových řad a ve zvolených hodnotách byla analýza rozšířena o výpočty s pomocí popisné statistiky.

Poté byly získané údaje ověřovány pomocí modifikací původního čistého provozu různými typy datových útoků a následným vyhodnocení, zda se anomálie podařilo či nepodařilo detekovat, a zda ze zjištěných údajů je možné odhadnout typ vedeného útoku.

Díky provedeným testům bylo ověřeno, že využití statistickým metod pro detekci anomálií má velký potenciál pro další výzkumy i pro reálné využití. Nemusí se jednat pouze o analýzu datového provozu. Tyto metody i postupy by mohly být využity i při testování různých zařízení v síti z pohledu komunikace v síti a zabezpečení. Ve většině případů se u takovýchto testů jedná o opakované testování stejných scénářů a sledování chování, která se porovnávají s předchozími měřeními.

Pro reálné nasazení těchto postupů je vždy důležitá detailní a dlouhodobá úvodní analýza, díky které je možné stanovit standardní hodnoty pro sledovaný provoz. Pomocí těchto hodnot lze nastavit hranice pro detekce anomálií ve sledovaných provozech. Pro další práci by bylo vhodné celý proces automatizovat, aby se záznamové a analytické skripty spouštěly v pravidelných časových intervalech a zároveň vytvořit propojení s datovým úložištěm, kam by získaná data mohla být přehledně ukládána pro snažší budoucí analýzu.

LITERATURA

[1] ARORA, Himanshu. TCP Attacks: TCP Sequence Number Prediction and TCP Reset Attacks. [online]. [cit. 2014-12-07]. Dostupné z: http://www.thegeekstuff.com/2012/01/tcp-sequence-number-attacks/

[2] BERNSTEIN, D. J. SYN cookies. [online]. [cit. 2014-12-06]. Dostupné z: http://cr.yp.to/syncookies.html

[3] BLAGOV, Maxim. UDP Flood. [online]. [cit. 2014-12-06]. Dostupné z: http://www.incapsula.com/ddos/attack-glossary/udp-flood.html

[4] ČMELÍK, Martin. Seznamte se – DoS a DDoS útoky. [online]. [cit. 2014-12-06]. Dostupné z: http://www.security-portal.cz/clanky/seznamte-se-%E2%80%93-dos-ddos-%C3%BAtoky

[5] ENDORF, Carl. *Detekce a prevence počítačového útoku*. 1. vyd. Praha: Grada, 2005, 355 s. ISBN 80-247-1035-8.

[6] GLYNN, Fergal. ARP Spoofing. [online]. [cit. 2014-12-06]. Dostupné z: http://www.veracode.com/security/arp-spoofing

[7] HALLER, Martin. Seriál Útoky typu DoS. *Lupa.cz* [online]. [cit. 2014-12-06]. Dostupné z: http://www.lupa.cz/serialy/utoky-typu-dos/#ic=serial-box&icc=title

[8] HELLER, Martin. Resetovací útoky na TCP spojení. [online]. [cit. 2014-12-06]. Dostupné z: http://www.lupa.cz/clanky/resetovaci-utoky-na-tcp-spojeni/

[9] HELLER, Martin. Jak unést TCP spojení. [online]. [cit. 2014-12-06]. Dostupné z: http://www.lupa.cz/clanky/jak-unest-tcp-spojeni/

[10] JANSSEN, Cory. Smurf Attack. [online]. [cit. 2014-12-06]. Dostupné z: http://www.techopedia.com/definition/17294/smurf-attack

[11] MATETI, Prabhaker. TCP Exploits. [online]. [cit. 2014-12-06]. Dostupné z: http://cecs.wright.edu/~pmateti/Courses/4420/TCPexploits/index.html

[12] MATETI, Prabhaker. Spoofing, Fragmentation, and Smurfing. [online]. [cit. 2014-12-09]. Dostupné z: http://cecs.wright.edu/~pmateti/InternetSecurity/Lectures/IPexploits/

[13] PISCITELLO, David. Anatomy of a DNS DDoS Amplification Attack. [online]. [cit. 2014-12-07]. Dostupné z: http://www.watchguard.com/infocenter/editorial/41649.asp

[14] PŘIBYL, Tomáš. Zákeřný útok jménem DoS. [online]. [cit. 2014-12-06]. Dostupné z: http://www.systemonline.cz/it-security/zakerny-utok-jmenem-dos.htm

[15] ROUSE, Margaret. Smurfing. In: [online]. [cit. 2014-12-06]. Dostupné z: http://searchsecurity.techtarget.com/definition/smurfing

[16] TANASE, Matthew. IP Spoofing: An Introduction. [online]. [cit. 2014-12-06]. Dostupné z: http://www.symantec.com/connect/articles/ip-spoofing-introduction

[17] WANG, Haining, Danlu ZHANG a Kang G. SHIN. Detecting SYN Flooding Attacks. [online]. [cit. 2014-12-09]. Dostupné z: http://www.cs.wm.edu/~hnw/paper/attack.pdf

[18] Buffer Overflow. [online]. [cit. 2014-12-08]. Dostupné z: https://www.owasp.org/index.php/Buffer_Overflow

[19] Method and system for UDP flood attack detection. [online]. [cit. 2014-12-09]. Dostupné z: http://www.google.com/patents/US8307430

[20] Kerio Control: Příručka administrátora. [online]. [cit. 2014-12-06]. Dostupné z: http://manuals.kerio.com/control/adminguide/cz/index.html

[21] Smurf. [online]. [cit. 2014-12-06]. Dostupné z: http://www.iss.net/security_center/advice/Exploits/IP/smurf/default.htm

[22] tshark [online]. [cit. 2015-05-16]. Dostupné z: https://www.wireshark.org/docs/man-pages/tshark.html

[23] HANČLOVÁ, Jana a Lubor TVRDÝ. Úvod do analýzy časových řad [online]. [cit. 2015-05-17]. Dostupné z: http://gis.vsb.cz/pan-old/Skoleni_Texty/TextySkoleni/AnalyzaCasRad.pdf

[24] Normální rozdělení [online]. [cit. 2015-05-18]. Dostupné z: http://www.wikiskripta.eu/index.php/Norm%C3%A1ln%C3%AD_rozd%C4%9Blen%C3%AD

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ACL	Access Control List
ARP	Address Resolution Protocol
csv	hodnoty oddělené čárkami – Comma-separated values
DAI	Dynamic ARP Inspection
DDoS	distribuované odmítnutí služby – Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DoS	odmítnutí služby – Denial of Service
EDNS	rozšiřující mechanizmy DNS – Extension mechanisms for DNS
FDS	Flooding Detection System
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDS	systém pro odhalení průniku – Intrusion Detection System
IP	Internet Protocol
LAN	místní síť – Local Area Network
MAC	Media Access Control
MTU	maximální přenosová jednotka – Maximum Transmission Unit
pcap	packet capture
SSH	Secure Shell
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol

SEZNAM PŘÍLOH

A POUŽITÉ SKRIPTY

A.1 Skript parse_den.ksh

```
#!/bin/bash

# usage:
# 1. run the script in directory with the pcap files for parsing
# 2. the results are in file parse.csv
# 3. the results file is deleted every time, in case it is already exists
# 4. there are no any testing, like empty file, wrong file etc.

export RESULTS="parse_den.csv"
export TOTAL
TOTAL=0

[ -f ./parse.csv ] && rm ./parse_den.csv
for PCAPFILE in `ls *.pcap`; do
  echo "doing: $PCAPFILE"
  TOTAL=`tshark -r $PCAPFILE -Y "frame.cap_len >= 0" | wc -l`
  echo "$PCAPFILE;$TOTAL" >> $RESULTS
TOTAL=0
done
```

A.2 Skript parse.ksh

```
#!/bin/bash

# usage:
# 1. run the script in directory with the pcap files for parsing
# 2. the results are in file parse.csv
# 3. the results file is deleted every time, in case it is already exists
# 4. there are no any testing, like empty file, wrong file etc.

export RESULTS="parse.csv"
export R300 R600 R900 R1200 R1500 R1501 TOTAL
R300=0
R600=0
R900=0
R1200=0
R1500=0
R1501=0
TOTAL=0

[ -f ./parse.csv ] && rm ./parse.csv
for PCAPFILE in `ls *.pcap`; do
  echo "doing: $PCAPFILE"
  R300=`tshark -r $PCAPFILE -Y "frame.len >= 0 &&
        frame.len <= 300" | wc -l`
  R600=`tshark -r $PCAPFILE -Y "frame.len >= 301 &&
        frame.len <= 600" | wc -l`
  R900=`tshark -r $PCAPFILE -Y "frame.len >= 601 &&
        frame.len <= 900" | wc -l`
  R1200=`tshark -r $PCAPFILE -Y "frame.len >= 901 &&
        frame.len <= 1200" | wc -l`
  R1500=`tshark -r $PCAPFILE -Y "frame.len >= 1201 &&
        frame.len <= 1500" | wc -l`
  R1501=`tshark -r $PCAPFILE -Y "frame.len >= 1501" | wc -l`
  TOTAL=`tshark -r $PCAPFILE -Y "frame.len" | wc -l`
  echo "$PCAPFILE;$R300;$R600;$R900;$R1200;$R1500;$R1501;$TOTAL"
    >> $RESULTS
R300=0; R600=0; R900=0; R1200=0; R1500=0; R1501=0; TOTAL=0
done
```

A.3 Skript flags.ksh

```
#!/bin/bash

# usage:
# 1. run the script in directory with the pcap file for TCP flag counting
# 2. the results are in file flags.csv
# 3. the results file is deleted every time, in case it is already exists
# 4. there are no any testing, like empty file, wrong file etc.

export RESULTS="flags.csv"
export SYN ACK SYNACK PUSHACK RST FIN
SYN=0
ACK=0
SYNACK=0
PUSHACK=0
RST=0
FIN=0

[ -f ./flags.csv ] && rm ./flags.csv

for PCAPFILE in `ls *.pcap`; do
  echo "doing: $PCAPFILE"
  SYN=`tshark -r $PCAPFILE -Y "tcp.flags.syn == 1 &&
        tcp.flags.ack == 0" | wc -l`
  ACK=`tshark -r $PCAPFILE -Y "tcp.flags.ack == 1" | wc -l`
  SYNACK=`tshark -r $PCAPFILE -Y "tcp.flags.syn == 1 &&
        tcp.flags.ack == 1" | wc -l`
  PUSHACK=`tshark -r $PCAPFILE -Y "tcp.flags.push == 1 &&
        tcp.flags.ack == 1" | wc -l`
  RST=`tshark -r $PCAPFILE -Y "tcp.flags.reset == 1" | wc -l`
  FIN=`tshark -r $PCAPFILE -Y "tcp.flags.fin == 1" | wc -l`
  echo "$PCAPFILE;$SYN;$ACK;$SYNACK;$PUSHACK;$RST;$FIN;" >> $RESULTS
  SYN=0; ACK=0; SYNACK=0; PUSHACK=0; RST=0; FIN=0
done
```

A.4 Skript ports.ksh

```
#!/bin/bash

# usage:
# 1. run the script in directory with the pcap files for counting ports
# 2. the results are in file ports.csv
# 3. the results file is deleted every time, in case it is already exists
# 4. there are no any testing, like empty file, wrong file etc.

export RESULTS="ports.csv"
export TCP22 UDP22 TCP25 UDP25 TCP53 UDP53 TCP80 UDP80 TCP110 UDP110
TCP143 UDP143 TCP443 UDP443 TCP465 UDP465 TCP993 UDP993 TCP995 UDP995
TCPWKNP UDPWKNP

TCP22=0 UDP22=0 TCP25=0 UDP25=0 TCP53=0 UDP53=0 TCP80=0 UDP80=0
TCP110=0 UDP110=0 TCP143=0 UDP143=0 TCP443=0 UDP443=0 TCP465=0
UDP465=0 TCP993=0 UDP993=0 TCP995=0 UDP995=0 TCPWKNP=0 UDPWKNP=0

[ -f ./ports.csv ] && rm ./ports.csv

for PCAPFILE in `ls *.pcap`; do
  echo "doing: $PCAPFILE"
  #SSH
  TCP22=`tshark -r $PCAPFILE -Y "tcp.port==22" | wc -l`
  UDP22=`tshark -r $PCAPFILE -Y "udp.port==22" | wc -l`
  #SMTP
  TCP25=`tshark -r $PCAPFILE -Y "tcp.port==25" | wc -l`
  UDP25=`tshark -r $PCAPFILE -Y "udp.port==25" | wc -l`
  #DNS
  TCP53=`tshark -r $PCAPFILE -Y "tcp.port==53" | wc -l`
  UDP53=`tshark -r $PCAPFILE -Y "udp.port==53" | wc -l`
  #HTTP
  TCP80=`tshark -r $PCAPFILE -Y "tcp.port==80" | wc -l`
  UDP80=`tshark -r $PCAPFILE -Y "udp.port==80" | wc -l`
  #POP3
  TCP110=`tshark -r $PCAPFILE -Y "tcp.port==110" | wc -l`
  UDP110=`tshark -r $PCAPFILE -Y "udp.port==110" | wc -l`
  #IMAP
```

```
  TCP143=‘tshark -r $PCAPFILE -Y "tcp.port==143" | wc -l‘
  UDP143=‘tshark -r $PCAPFILE -Y "udp.port==143" | wc -l‘
  #HTTPS
  TCP443=‘tshark -r $PCAPFILE -Y "tcp.port==443" | wc -l‘
  UDP443=‘tshark -r $PCAPFILE -Y "udp.port==443" | wc -l‘
  #SMTPS
  TCP465=‘tshark -r $PCAPFILE -Y "tcp.port==465" | wc -l‘
  UDP465=‘tshark -r $PCAPFILE -Y "udp.port==465" | wc -l‘
  #IMAPS
  TCP993=‘tshark -r $PCAPFILE -Y "tcp.port==993" | wc -l‘
  UDP993=‘tshark -r $PCAPFILE -Y "udp.port==993" | wc -l‘
  #POP3S
  TCP995=‘tshark -r $PCAPFILE -Y "tcp.port==995" | wc -l‘
  UDP995=‘tshark -r $PCAPFILE -Y "udp.port==995" | wc -l‘
  #WellKnownPorts
  TCPWKNP=‘tshark -r $PCAPFILE -Y "tcp.port>=0 &&
        tcp.port<=1023" | wc -l‘
  UDPWKNP=‘tshark -r $PCAPFILE -Y "udp.port>=0 &&
        udp.port<=1023" | wc -l‘

echo "$PCAPFILE;$TCP22;$UDP22;$TCP25;$UDP25;$TCP53;$UDP53;$TCP80;
    $UDP80;$TCP110;$UDP110;$TCP143;$UDP143;$TCP443;$UDP443;$TCP465;
    $UDP465;$TCP993;$UDP993;$TCP995;$UDP995;$TCPWKNP;$UDPWKNP"
    >> $RESULTS

TCP22=0;UDP22=0;TCP25=0;UDP25=0;TCP53=0;UDP53=0;TCP80=0;UDP80=0;
TCP110=0;UDP110=0;TCP143=0;UDP143=0;TCP443=0;UDP443=0;TCP465=0;
UDP465=0;TCP993=0;UDP993=0;TCP995=0;UDP995=0;TCPWKNP=0;UDPWKNP=0

done
```

A.5 Skript protocol.ksh

```
#!/bin/bash

# usage:
# 1. run the script in directory with the pcap files for
     sellected protocol counting
# 2. the results are in file protocol.csv
# 3. the results file is deleted every time, in case it is already exists
# 4. there are no any testing, like empty file, wrong file etc.

export RESULTS="protocol.csv"
export TCP HTTP FTP UDP DNS ICMP
TCP=0
HTTP=0
FTP=0
UDP=0
DNS=0
ICMP=0

[ -f ./protocol.csv ] && rm ./protocol.csv

for PCAPFILE in `ls *.pcap`; do
  echo "doing: $PCAPFILE"
  TCP=`tshark -r $PCAPFILE -Y "tcp" | wc -l`
  HTTP=`tshark -r $PCAPFILE -Y "http" | wc -l`
  FTP=`tshark -r $PCAPFILE -Y "ftp" | wc -l`
  UDP=`tshark -r $PCAPFILE -Y "udp" | wc -l`
  DNS=`tshark -r $PCAPFILE -Y "dns" | wc -l`
  ICMP=`tshark -r $PCAPFILE -Y "icmp" | wc -l`
  echo "$PCAPFILE;$TCP;$HTTP;$FTP;$UDP;$DNS;$ICMP" >> $RESULTS
  TCP=0; HTTP=0; FTP=0; UDP=0; DNS=0; ICMP=0
done
```

A.6 Skript ports_an.ksh

```
#!/bin/bash

# usage:
# 1. run the script in directory with the pcap files for anaylizng
# 2. the results will be written on display
# 3. the results file is deleted every time, in case it is already exists
# 4. there are no any testing, like empty file, wrong file etc.

#zavedení hranic pro žlutý a červený alarm, hodnota daná v %, bash umí
pracovat pouze s celočíselnými proměnnými

base=95
yellow=85
red=75

#sečteme celkový provoz na WKnPortech 0-1023

T=‘tshark -r a.pcap -Y "(tcp.port>=0 or udp.port>=0) &&
        (tcp.port<=1023 or udp.port<=1023)" | wc -l‘
echo "Celkový počet sledovaných paketů je" $T

#zároveň si tento provoz uložíme do souboru b.pcap pro zrychlení chodu

tshark -r a.pcap -Y "(tcp.port>=0 or udp.port>=0) &&
        (tcp.port<=1023 or udp.port<=1023)" -w b.pcap

#budeme sčítat provoz na portech daných v souboru porty.txt,
default je 25, 53, 80, 143, 443, 465, 993

while read radek; do
   P=‘tshark -r b.pcap -Y "tcp.port==$radek or udp.port==$radek" | wc -l‘
   #echo $radek
   x=$[$predchozi+$P]
   predchozi=$x
   #echo $x;
   done < porty.txt
```

```
#spočteme poměr provozu na portech daných v souboru porty.txt a
celkovým provozem

    pomer=$[$x * 100 / $T]
    echo "V analyzovaném provozu je $pomer % sledovaných paketů."

#porovnáme s danou vstupní hodnotou - získanou z úvodní analýzy
průměr-odchylka (přepoklad 95%+-)
#pokud bude poměr menší než tato hodnota, zobrazí žlutý alarm
#jestliže je poměr menší než průměr-2*odchylka, potom zobrazí
red alarm

if [ "$pomer" -lt "$red" ] ;
   then echo -e "\e[31mRed Alarm!!! \e[39mKontrolovaná hodnota poměru
   je výrazně pod sledovaným limitem."
    echo "Pro bližší detaily otevřte soubor b.pcap ve Wiresharku a spusťte
        Statistics > Converstaions."
    elif [ "$pomer" -lt "$yellow" ] ;
   then echo -e "\e[33mYellow Alarm!!! \e[39mKontrolovaná hodnota poměru
   je pod sledovaným limitem."
else
   echo "Kontrola dokončena. Naměřené hodnoty jsou v pořádku."
fi
done
```

A.7 Skript parse_an.ksh

```
#!/bin/bash

# usage:
# 1. run the script in directory with the pcap files for anaylizng
# 2. the results will be written on display
# 3. the results file is deleted every time, in case it is already exists
# 4. there are no any testing, like empty file, wrong file etc.

#deklarace proměnných získaných z analýzy provozu a výpočtů,
zaokoruhleno na celá čísla

avg=30 #průměr
 dev=5 #odchylka

#nastavení hranic alarmů pro dané hodnoty, v případě změny
charakteristiky provozu stačí upravit pouze průměr a odchylku

yeld=$[$avg - $dev]
yelu=$[$avg + $dev]
redd=$[$yeld - $dev]
redu=$[$yelu + $dev]

#celkový počet paketů
T=‘tshark -r a.pcap -Y "frame.len" | wc -l‘
echo "Celkový počet sledovaných paketů je" $T

#počet paketů o sledované velikosti
P=‘tshark -r a.pcap -Y "frame.len >= 0 && frame.len <= 300" | wc -l‘

#poměr sledovaných paketů s celkovým provozem
pomer=$[$P * 100 / $T]
echo "V analyzovaném provozu je $pomer % sledovaných paketů."

#porovnáme se vstupními hodnotami
if [ "$pomer" -lt "$redd" ] || [ "$pomer" -gt "$redu" ];
    then echo -e "\e[31mRed Alarm!!! \e[39mV analyzovaném provozu je
        detekována výrazná odlišnost výskytu paketů o velikosti 0-300 kB
```

```
        oproti standardnímu provozu."
elif [ "$pomer" -lt "$yeld" ] || [ "$pomer" -gt "$yelu" ];
    then echo -e "\e[33mYellow Alarm!!! \e[39mV analyzovaném provozu je
        detekována odlišnost výskytu paketů o velikosti 0-300 kB oproti
        standardnímu provozu."
else
    echo "Kontrola dokončena. Naměřené hodnoty jsou v pořádku."
fi

echo "Standardní hodnoty by se měli pohybovat mezi $yeld a $yelu %."
done
```

OBSAH PŘILOŽENÉHO DVD

- Soubor Diplomova_prace_xwoidi00_165848.pdf – elektronická verze diplomové práce.
- Adresář Časové řady – sešity aplikace MS Excel s řešením analýz s pomocí časových řad.
- Adresář Obrázky - snímky grafů, tabulek a dalších obrázků použitých v práci.
- Adresář Popisná statistika – sešity aplikace MS Excel s řešením analýz s pomocí popisné statistiky.
- Adresář Testovací analýza – sešity aplikace MS Excel s řešením testovací analýzy a použité *pcap* soubory. Obsahuje 3 podadresáře:
  - DNS – soubory související s útokem DNS,
  - Ostinato – provoz vygenerovaný pomocí aplikace Ostinato a související soubory,
  - SYN flood – SYN flood útok a s ním související soubory.
- Adresář Zdrojová data - *csv* soubory s daty ze skriptů a sešity MS Excel se souhrnnými daty za všechny dny a za pracovní dny pro všechny sondy, na kterých byl prováděn záznam.